ŚLĄSKA WYŻSZA SZKOŁA INFORMATYCZNO-MEDYCZNA

WYDZIAŁ GRAFIKI I INFORMATYKI

KIERUNEK: INFORMATYKA

ADAM MICHAŁ ZIAJA

MAPY SIECI BEZPRZEWODOWYCH

Praca dyplomowa napisana pod kierunkiem

dr inż. Paweł Kasprowski

……..
(podpis promotora)

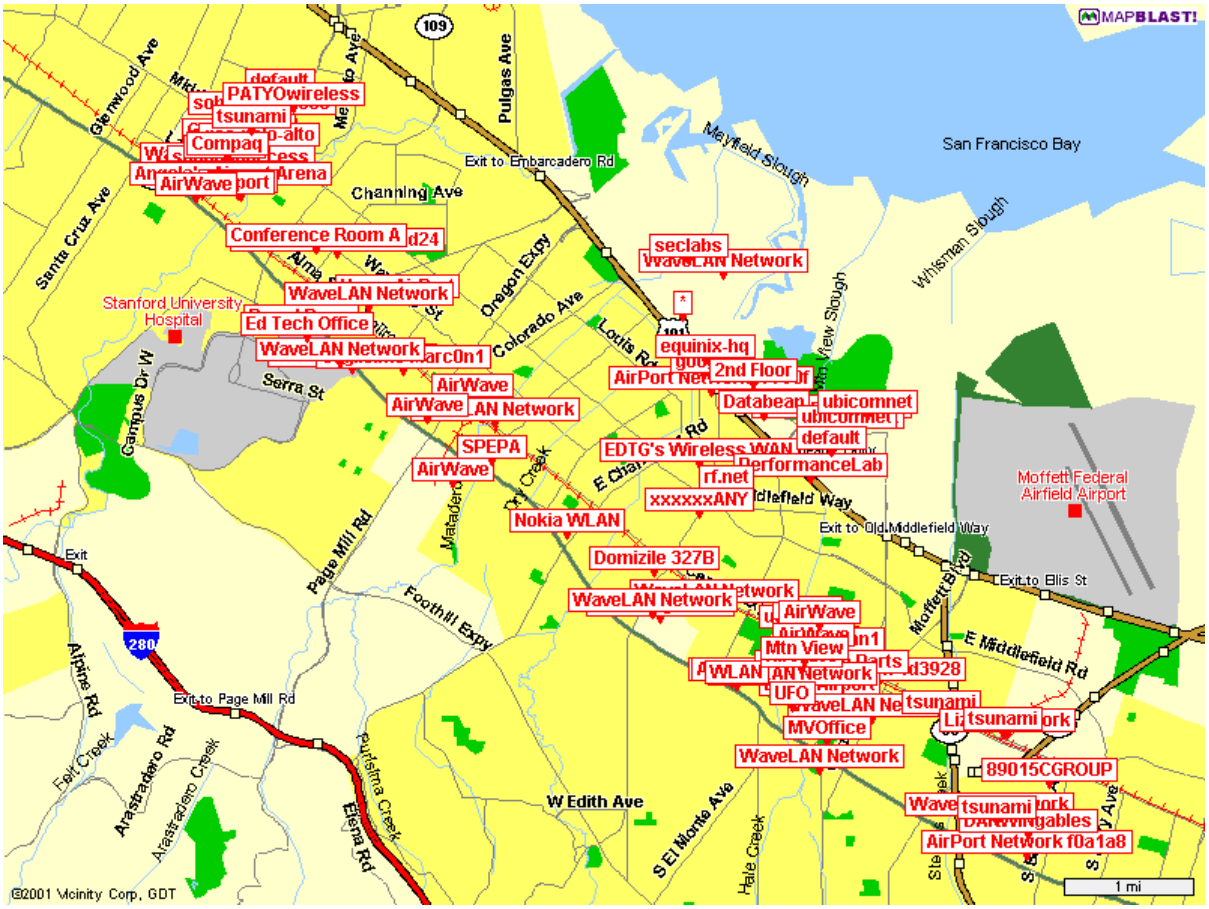
CHORZÓW 2011

Spis treści

1. Wstęp

Sieci bezprzewodowe szczególnie na przestrzeni ostatnich lat stały się bardzo powszechne. Moja praca „mapy sieci bezprzewodowych" związana jest głównie z pojęciami wardriving oraz warchalking.

Pojęcie wardriving oznacza wyszukiwanie miejsc w których dostępne są sieci bezprzewodowe przez osobę poruszającą się pojazdem z przenośnym komputerem lub innym urządzeniem umożliwiającym wyszukiwanie bezprzewodowych sieci, takim jak np. PDA czy Smartphone. Najbardziej popularnym oprogramowaniem do wardrivingu jest Kismet pod system operacyjny Linux oraz NetStumbler pod system Windows. Wardriving wymyślił Peter Shipley w okolicach lat 1999-2000, który jako pierwszy zautomatyzował cały proces z dedykowanym oprogramowaniem oraz odbiornikiem GPS, który w momencie znalezienia sieci zapisywał pozycję z GPS, co pozwoliło mu na stworzenie pierwszych map sieci bezprzewodowych.

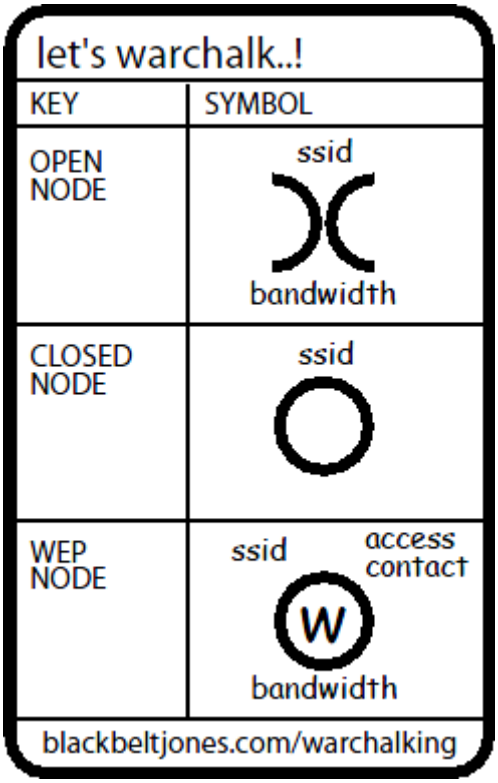


Mapa z oznaczonymi nazwami sieci stworzona przez Peter Shipley.

Źródło: http://www.dis.org/wl/maps/

Nazwa wardriving pochodzi od słowa wardialing, techniki spopularyzowanej przez postać graną przez Matthew Broderick w filmie WarGames, która polega na wykorzystywaniu komputera do wybierania numerów telefonów w nadziei na znalezienie aktywnego modemu.

Drugim pojęciem, z jakim związana jest moja praca jest warchalking. Warchalking wymyślił Matt Jones w 2002 roku, pojęcie oznacza oznaczanie miejsc, w których dostępne są sieci bezprzewodowe. Pojęcie powstało przez analogię do słowa wardriving. Pierwotnym założeniem było oznaczanie miejsc po przez rysowanie kredą specjalnych symboli w miejscu, w którym wykryto siec bezprzewodową. Symbole wykorzystywane w warchalkingu inspirowane były symbolami hobo.



Karta z symbolami wykorzystywanymi w warchalkingu stworzona przez Matt Jones.

Źródło: http://www.blackbeltjones.com/warchalking/warchalking0_9.pdf

Tematem pracy są „mapy sieci bezprzewodowych" głównie z powodu, że wardriving oraz warchalking są bardzo mało popularne w Polsce, praktycznie nie istnieje żadna polskojęzyczna literatura opisująca te interesujące społeczne zjawiska. Oba te pojęcia wraz z rozwojem technologii ewoluowały i dziś sprowadzają się głównie do oznaczania znalezionych sieci bezprzewodowych na interaktywnych mapach internetowych z

wykorzystaniem technologii takich jak Google Maps API czy OpenLayers API, które to są otwartym oprogramowaniem (ang. open source). Moja praca jest dość istotna z punktu widzenia dzisiejszego bezpieczeństwa informacji, które przepływają przez sieci bezprzewodowe, ponieważ to praktycznie jedyna metoda, która pozwala na powszechny audyt bezpieczeństwa sieci bezprzewodowych po przez wykrycie szyfrowania w nich stosowanego. Szyfrowanie WEP zostało złamane w roku 2001, jednak do dnia dzisiejszego jest powszechnie stosowane, m.in. w jednej z najbardziej popularnych usług Internetu bezprzewodowego w naszym kraju czyli usługi Livebox TP, oferowanej przez Telekomunikację Polską. Przy czym również znaczna część sieci nie jest w ogóle szyfrowana.

Jako narzędzia do badań wykorzystałem netbooka Asus Eee PC 1000H z kartami Wi-Fi RaLink RT2860 i Sagem XG-76NA 802.11bg (chipset ZyDAS ZD1211) oraz odbiornik GPS Holux M-241 (podłączony przez Bluetooth), a jako oprogramowanie udostępniający najwięcej informacji o bezprzewodowych sieciach lokalnych program Kismet, pracujący pod kontrolą darmowego systemu operacyjnego Debian GNU/Linux, jako środek transportu posłużył mi samochód osobowy.

Mam nadzieję, że niniejsza praca jako swojego rodzaju audyt bezpieczeństwa przyczyni się do zwrócenia uwagi społeczeństwa na aspekty zabezpieczenia informacji przepływających przez bardzo słabo zabezpieczone sieci bezprzewodowe, a w związku z tym do poprawy ich bezpieczeństwa.

2. Ogólna charakterystyka sieci bezprzewodowych

Pierwsza bezprzewodowa sieć komputerowa została stworzona na przełomie lat 60 i 70 na Uniwersytecie Hawajskim przez zespół badaczy pod przywództwem Normana Abramsona. Była nią ALOHAnet znana również jako ALOHA System. Jednak pierwszy standard Wi-Fi pojawił się na rynku dopiero w 1999 roku, był nim 802.11b, a następnie rok później 802.11a, chodź oba te standardy zostały wydane w tym samym czasie. W 2003 roku organizacja IEEE (Instytut Inżynierów Elektryków i Elektroników, od ang. Institute of Electrical and Electronics Engineers) opracowała nowszą specyfikację w postaci standardu 802.11g, który to był połączeniem najlepszych cech standardów 802.11b i 802.11a. Standard 802.11g oferuje tą samą szybkość co standard 802.11a oraz większy zasięg niż 802.11b. Specyfikacja 802.11g jest zgodna ze standardem 802.11b, co za tym idzie, karta sieciowa 802.11b będzie działać z punktem dostępu 802.11g, jednak wolniej, z prędkością 802.11b.

Najnowszą specyfikacją Wi-Fi jest 802.11n, nad którą prace zostały ukończone w drugiej połowie 2009 roku. Oferuje ona przede wszystkim większą szybkość w porównaniu do poprzednich wersji standardów z grupy 802.11, nawet do 300 Mb/s, czyli trzy razy więcej niż Fast Ethernet. Punkty dostępów w technologii 802.11n działają również z kartami sieciowymi 802.11b oraz 802.11g, jednak z prędkością starszego standardu.

Sieci bezprzewodowe naturalnie mają swoje wady i zalety, do zalet należą przede wszystkim:

- łatwa budowa oraz rozbudowa sieci pozbawionej kabli;
- możliwość korzystania z bezprzewodowego Internetu poprzez lokalnych dostawców, bez potrzeby doprowadzania okablowania;
- darmowy dostęp do Internetu po przez różnego rodzaju HotSpoty;
- swoboda i mobilność – bezprzewodowe podłączenie do sieci urządzeń mobilnych takich jak telefony czy palmtopy;
- łatwo dostępne i coraz to tańsze urządzenia Wi-Fi;
- możliwość podłączenia większej ilości urządzeń z mniejszym nakładem kosztów, domowe routery Wi-Fi pozwalają podłączyć nawet około 250 klientów;
- duża odporność na wyładowania atmosferyczne w porównaniu do sieci kablowych;
- możliwość łączenia się z Internetem nawet w ruchu;
- tanie i szybkie w instalacji;

- możliwość stosowania w obiektach zabytkowych, gdzie nie można stosować okablowania;

oczywiście sieci bezprzewodowe mają również swoje wady, należą do nich przede wszystkim:

- bardzo podatne na zakłócenia;
- urządzenie Wi-Fi musi być poprawnie skonfigurowane, w przeciwnym wypadku może być łatwym celem na różnego typu ataki, takie jak np. przechwytywanie danych w sieciach WLAN, czy sterowanie urządzeniami np. typu kamery IP;
- połączenia na dalekie odległości mogą okazać się niestabilne, gdy sygnał z punktu dostępowego będzie zbyt słaby;
- sieci bezprzewodowe są mniej bezpieczne od kablowych, przez co konieczne jest stosowanie dodatkowych zabezpieczeń, które w efekcie zmniejszają prędkość przesyłu danych;
- zazwyczaj w miastach, szczególnie na blokowiskach, sieci wzajemnie się zagłuszają;
- szybkość transmisji zależna jest od odległości między urządzeniami komunikującymi się za pomocą Wi-Fi;
- standardy Wi-Fi 802.11b, 802.11g oraz 802.11n wykorzystują pasmo 2,4 GHz, w tym samym zakresie działa wiele urządzeń, takie jak kuchenki mikrofalowe, telefony bezprzewodowe, radiowa telewizja przemysłowa, urządzenia Bluetooth i wiele innych. W związku z tym mogą się wzajemnie zagłuszać i w efekcie ograniczać zasięg sygnału Wi-Fi;

Zgodnie z regulacjami międzynarodowymi zakres częstotliwości radiowych wokół częstotliwości 2,4 GHz nie wymagają koncesji i jest przeznaczony do zastosowań przemysłowych, naukowych i medycznych (są to tzw. pasma ISM, od ang. Industrial, Scientific & Medical), oraz dla sieci bezprzewodowych, działających w systemie rozproszonego widma. Z tych pasm korzystają wszystkie aktualne standardy Wi-Fi czyli 802.11b, 802.11g oraz 802.11n.

Zakres częstotliwości w okolicach 5.3 GHz określa się jako pasma U-NII (ang. The Unlicensed National Information Infrastructure). FCC (Federalna Komisja Łączności, od ang. The Federal Communications Commission) w USA i inne podobne urzędy regulacyjne w innych krajach (w tym w Polsce) dopuszczają korzystanie z sieci bezprzewodowych zarówno

w pasmach ISM, jak i U-NII. Z pasma U-NII korzystają wszystkie sieci bezprzewodowe w standardzie 802.11a.

Dokładne wartości przydzielonych częstotliwości z zakresu 2,4 GHz są nieco inne od siebie w różnych zakątkach świata, mianowicie:

- Europa – zakres częstotliwości od 2,4000 do 2,4835 GHz;
- Ameryka Północna – zakres częstotliwości od 2,4000 do 2,4835 GHz;
- Hiszpania – zakres częstotliwości od 2,445 do 2,475 GHz;
- Francja – zakres częstotliwości od 2,4465 do 2,4835 GHz;
- Japonia – zakres częstotliwości od 2,471 do 2,497 GHz;

Praktycznie każde państwo na świecie wykorzystuje jedno z tych pasm. Nieznaczne różnice w przydziale częstotliwości nie mają większego znaczenia, a zakresy częstotliwości przyjęte w różnych krajach nakładają się w znacznym stopniu na siebie, co umożliwia wykorzystanie tego samego sprzętu Wi-Fi w różnych krajach. Producenci urządzeń sieciowych zazwyczaj ustawiają w nich zestaw kanałów przeznaczony dla kraju, w którym ten sprzęt będzie sprzedawany.

Częstotliwość działania sieci Wi-Fi zależy od wykorzystywanego kanału, na szczęście, na całym świecie używa się tej samej numeracji kanałów, w związku z tym kanał numer 11 w Warszawie zajmuje dokładnie tę samą częstotliwość co kanał 11 w Berlinie czy w Waszyngtonie. Lista kanałów wraz z częstotliwościami i regionem w którym mogą być wykorzystywane prezentuje się następująco:

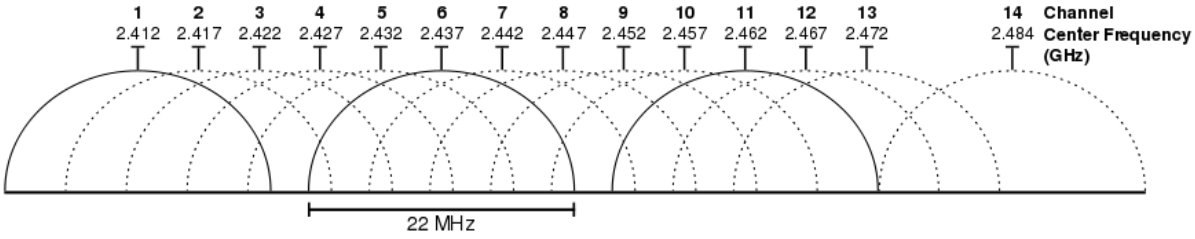
- Kanał 1 – 2,412 GHz – USA, Kanada, EMEA (kraje leżące na obszarze Europy, Bliskiego Wschodu oraz Afryki, od ang. Europe, the Middle East and Africa), Chiny, Japonia;
- Kanał 2 – 2,417 GHz – USA, Kanada, EMEA, Chiny, Japonia;
- Kanał 3 – 2,422 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;
- Kanał 4 – 2,427 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;
- Kanał 5 – 2,432 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;
- Kanał 6 – 2,437 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;
- Kanał 7 – 2,442 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;
- Kanał 8 – 2,447 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;
- Kanał 9 – 2,452 GHz – USA, Kanada, EMEA, Izrael, Chiny, Japonia;

- Kanał 10 – 2,457 GHz – USA, Kanada, EMEA, Francja, Chiny, Japonia;
- Kanał 11 – 2,462 GHz – USA, Kanada, EMEA, Francja, Chiny, Japonia;
- Kanał 12 – 2,467 GHz – EMEA, Francja, Japonia;
- Kanał 13 – 2,472 GHz – EMEA, Francja, Japonia;
- Kanał 14 – 2,484 GHz – Japonia;

We Francji, mimo, że leży ona na terenie Europy, można korzystać tylko z kanałów 10, 11, 12, 13 oraz 14, czyli jedynie z zakresu częstotliwości od 2,457 do 2,484 GHz. Dla porównania w Polsce można korzystać z zakresu częstotliwości od 2,4000 do 2,4835 GHz, co odpowiada zakresowi kanałów Wi-Fi od 1 do 13, kanał 14, który działa na częstotliwości 2,484 GHz wymaga w Polsce koncesji. Na całym świecie poza Izraelem dostępne są kanały 10 i 11, przez co są one jednymi z najczęściej wybieranych standardowo przez producentów sprzętu Wi-Fi.

Częstotliwości odpowiadające każdemu z tych kanałów w rzeczywistości są częstotliwościami środkowymi dla pasm o szerokości 22 MHz. Dlatego też każdy z kanałów Wi-Fi nakłada się z kilkoma innymi, położonymi wyżej lub niżej. W całym paśmie 2,4 GHz występują tylko trzy całkowicie nie nakładające się kanały, są nimi 1, 6, 11. W związku z tym, jeśli np. w bloku mieszkalnym będą znajdować się blisko siebie sieci na kanałach 4, 5, 6 to będą wzajemnie się zakłócać, oczywiście wszystkie sieci będą działać, jednak ich wydajność, czyli szybkość transferu danych oraz stabilność połączenia nie będzie tak dobra jak w przypadku, gdy sieci będą działać przykładowo na kanałach 1, 6 oraz 11.

Dzisiejsze urządzenia takie jak routery bezprzewodowe posiadają możliwość wybrania automatycznego kanału, przez co urządzenie samo wybiera kanał na którym będzie najmniej zakłócać się z innymi sieciami Wi-Fi w zasięgu. Jeśli to możliwe, każda z sieci powinna korzystać z kanałów, które są oddalone od siebie przynajmniej o 25 MHz lub o pięć numerów, w przypadku kanałów od 1 do 13, ponieważ kanał 14 zakłócają tylko kanały 12 oraz 13.



Graficzne przedstawienie kanałów Wi-Fi w paśmie 2,4 GHz.

Źródło: http://en.wikipedia.org

Jak już wcześniej wspomniałem, mimo, że sieci będą się zakłócać to będą działać, jednak spadnie ich wydajność, natomiast większym problemem ze względu na zakłócenia od nakładających się wzajemnie kanałów Wi-Fi są inne urządzenia korzystające z pasma 2,4 GHz, takie jak np. telefony bezprzewodowe czy kuchenki mikrofalowe, które to muszą również korzystać z tego pasma ponieważ płynna woda, z powodu bardzo silnego tłumienia drgań nie posiada wyraźnego maksimum rezonansowego, lecz pochłania silnie fale elektromagnetyczne w dość szerokim zakresie częstotliwości mikrofalowych. Przy częstotliwości 2,45 GHz cząstki wody drgają na tyle szybko, by zapewnić dobre pochłanianie, a tym samym szybkie ogrzewanie potrawy. Częstotliwość mikrofal kuchenki musi mieścić się tym zakresie i jest wynikiem kompromisu pomiędzy dostępnymi częstotliwościami w paśmie radiowym ISM, a głębokością wnikania fal.

Specyfikacja standardu Wi-Fi 802.11a używa innego zakresu częstotliwości radiowych niżeli standardy 802.11b/g/n. W standardzie 802.11a kanały mają szerokość 20 MHz, mają tu więc zastosowanie praktycznie te same reguły dotyczące ich rozdzielania, ponieważ w specyfikacjach 802.11b/g/n kanały mają szerokość 22 MHz. Częstotliwości radiowe w sieciach 802.11a wraz z regionem w którym mogą być wykorzystywane prezentują się następująco:

- Kanał 34 – 5,17 GHz – Japonia;
- Kanał 36 – 5,18 GHz – Ameryka Północna, Europa, Singapur;
- Kanał 38 – 5,19 GHz – Japonia;
- Kanał 40 – 5,20 GHz – Ameryka Północna, Europa, Singapur;
- Kanał 42 – 5,21 GHz – Japonia;
- Kanał 44 – 5,22 GHz – Ameryka Północna, Europa, Singapur;
- Kanał 46 – 5,23 GHz – Japonia;

- Kanał 48 – 5,24 GHz – Ameryka Północna, Europa, Singapur;
- Kanał 52 – 5,26 GHz – Ameryka Północna, Tajwan;
- Kanał 56 – 5,28 GHz – Ameryka Północna, Tajwan;
- Kanał 60 – 5,30 GHz – Ameryka Północna, Tajwan;
- Kanał 64 – 5,32 GHz – Ameryka Północna, Tajwan;

Specyfikacje 802.11 oraz różne narodowe urzędy regulacyjne, takie jak Federalna Komisja Łączności w USA czy Urząd Komunikacji Elektronicznej w Polsce nakładają ponadto ograniczenia na maksymalną dopuszczalną moc nadajników oraz zysk anten, których można używać w sieciach bezprzewodowych Wi-Fi. Ograniczenia te mają naturalnie na celu zmniejszenie zasięgu działania sieci bezprzewodowej, aby na tym samym kanale mogła działać większa liczba sieci, które nie będą wzajemnie się zakłócać, ponieważ zasięg przeciętnego routera Wi-Fi to kilkadziesiąt metrów. Wpływ na zasięg sieci mają oczywiście napotkane przez sygnał przeszkody, takie jak chodź by ściany w mieszkaniu.

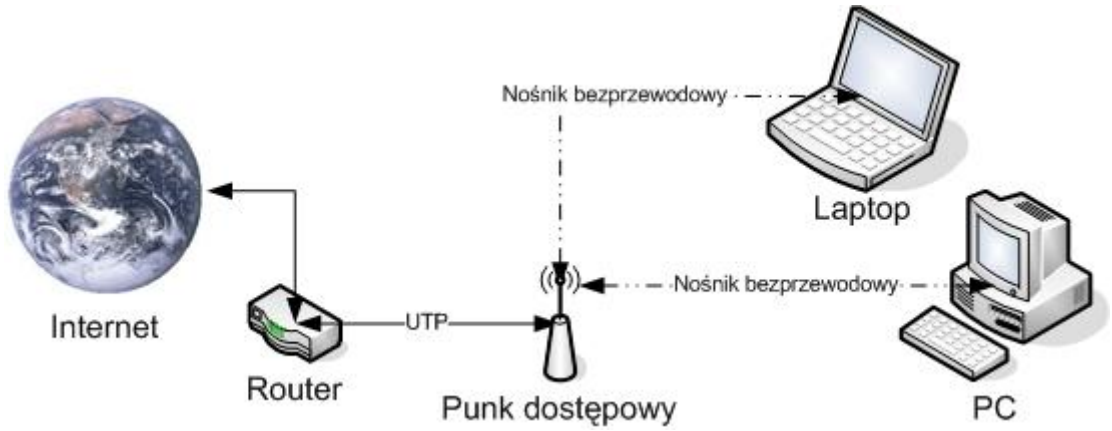
W sieciach bezprzewodowych wyróżniamy głównie dwa rodzaje urządzeń bezprzewodowych, są nimi karty sieciowe oraz punkty dostępu. Karta sieciowa jest podzespołem komputera lub innego urządzenia (np. telefon komórkowy), które wymienia dane za pośrednictwem sieci. Punkt dostępu (od ang. Access Point) to stacja bazowa sieci bezprzewodowej lub router łączący w tym wypadku sieć bezprzewodową z inna siecią komputerową, w tym o tradycyjnej przewodowej (kablowej) strukturze.

Karty sieciowe przeznaczone do bezprzewodowych sieci lokalnych mogą występować w kilku różnych wersjach, ze względu na metodę podłączenia do komputera lub innego urządzenia. Wyróżnia się więc głównie karty sieciowe:

- PCMCIA, podłączane do gniazda PCMCIA w laptopach, karty te zazwyczaj wystają parę centymetrów z gniazda aby obejść ograniczenie związane z wewnętrznym ekranowaniem. Część kart PCMCIA ma specjalne gniazdo przeznaczone do podłączenia anteny zewnętrznej w celu zwiększenia sygnału;
- PCI, umieszczane w złączu na płycie głównej stacjonarnego komputera;
- USB, podłączane do portu USB, który to zazwyczaj występuje w każdym typie komputera, od stacjonarnych po netbooki, a nawet komputery typu plug;

- Wewnętrzne, wbudowane karty sieciowe, najczęściej montowane w urządzeniach przenośnych takich jak laptopy czy telefony komórkowe, ale również drukarki, aparaty cyfrowe, radia internetowe, telefony VoIP itd.

Punkt dostępu (od ang. Access Point, AP) jest to urządzenie sieciowe pozwalające na połączenie ze sobą wielu innych urządzeń sieciowych, takich jak np. różnego rodzaju komputery, drukarki sieciowe itd. Punkty dostępu mogą komunikować się między sobą, co pozwala tworzyć bardzo rozległe sieci bezprzewodowe. Większość aktualnie produkowanych urządzeń typu Access Point posiada wbudowany router z modemem, który umożliwia tworzenie sieci mieszanych, czyli takich, które wykorzystująca więcej niż jedną technologię sieciową np. sieć bezprzewodowa i przewodową. Zależnie od rodzaju urządzenia punkty dostępu mogą mieć wiele innych użytecznych funkcji, zazwyczaj są nimi serwer DHCP, serwer DNS czy umiejętność translacji adresów prywatnych na publiczne, czyli NAT. Najlepszym przykładem punktu dostępowego z wbudowanym routerem oraz modemem, a także posiadającym wyżej wymienione funkcje jest popularny Livebox. Nowsze urządzenia Access Point posiadają również m.in. serwer plików oraz serwer FTP, a także umożliwiają podłączenie innych urządzeń jak np. dyski twarde na USB, oraz mogą również działać jako regenerator sygnału (od ang. repeater), przykładem takiego wielofunkcyjnego bezprzewodowego routera jest chodź by Netgear DGN2200.



Ilustracja działania punktu dostępowego.

Źródło: http://pl.wikipedia.org

Punkt dostępu jak każde bezprzewodowe urządzenie ma ograniczony zasięg, wynika to nie tylko z aspektów prawnych, ale również z kwestii technicznych, w części modeli jest jednak możliwość zwiększenia mocy po przez zewnętrzną antenę. Na zasięg punktu

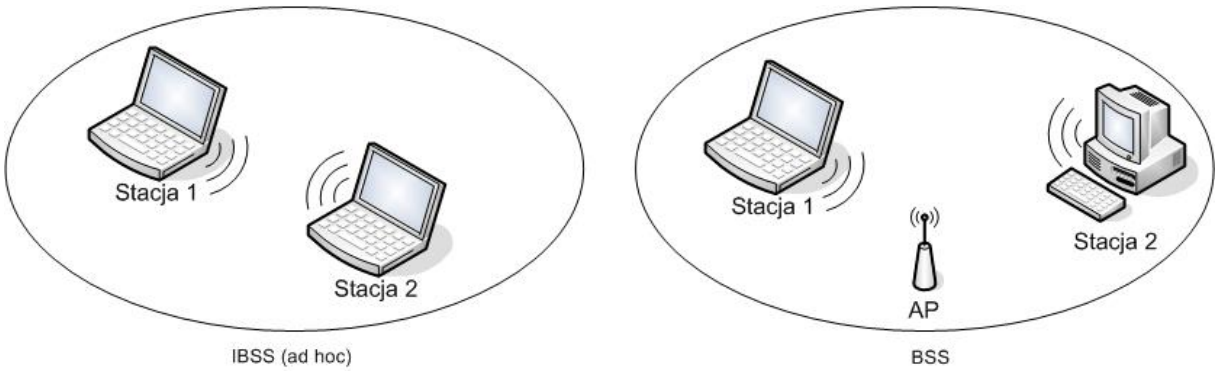
dostępowego poza mocą użytej anteny ma wpływ także umiejscowienie, czyli rodzaj przeszkód w okolicy, jak również inne elektroniczne urządzenia działające na tej samej częstotliwości jak np. mikrofalówki czy telefony bezprzewodowe, a dla urządzeń znajdujących się na otwartej przestrzeni także warunki atmosferyczne, szczególnie zimą, kiedy to anteny ulegają oblodzeniu.

Za pomocą punktów dostępowych możemy stworzyć następujący typy sieci bezprzewodowych:

- Sieć typu BSS (podstawowy zestaw usługowy, od ang. Basic Service Set),

ten typ sieci pozwala na przesyłanie danych między klientami za pośrednictwem punkty dostępowego, w związku z tym, wszyscy klienci muszą być w zasięgu urządzenia. W standardzie nie istnieje ograniczenie ilości stacji podłączonych do jednego punktu dostępowego, jednak niska przepustowość sieci bezprzewodowych, jak i również wydajność urządzenia wymagają ograniczenia liczby klientów. Każdy BSS ma unikatowy 48-bitowy identyfikator BSSID (od ang. Basic Service Set Identifier) i jest on zazwyczaj adresem MAC bezprzewodowego interfejsu punktu dostępowego.

Istnieje również siec typu BBS bez punktu dostępowego, stacje w takiej sieci komunikują się ze sobą bezpośrednio, a sieć taka nosi nazwę IBSS (niezależny BSS, od ang. independent BSS) , jest również nazywa siecią tymczasową lub ad-hoc.

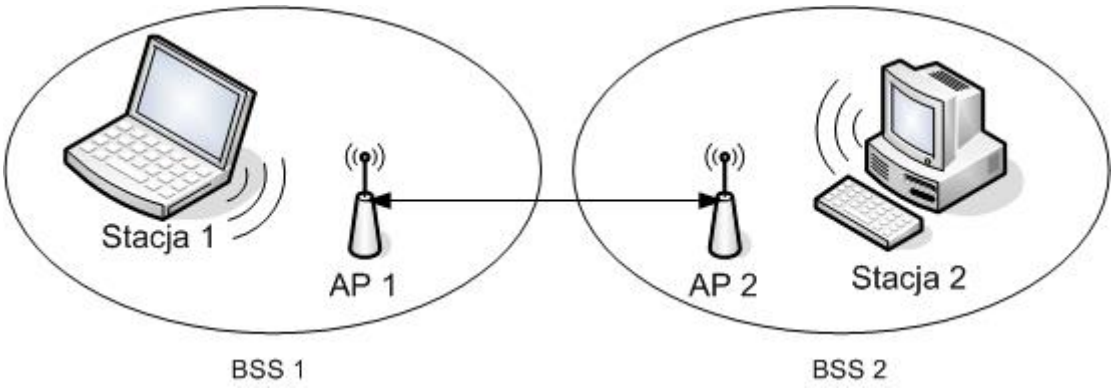


Różne typy sieci BSS.

Źródło: http://pl.wikipedia.org

- Sieć typu ESS (rozszerzony zestaw usług, od ang. Extended Service Set),

ten typ sieci natomiast powstaje w wyniku połączenia kilku BSS za pomocą sieci szkieletowej. W takiej sieci klienci mogą przemieszczać się bez utraty połączenia pomiędzy różnymi BSS, pod warunkiem, że wszystkie są częścią tego samego ESS. Model takiej sieci najczęściej wykorzystywany jest do tworzenia hotspotów ("gorący punkt", od ang. hot spot), czyli publicznie dostępnych sieci bezprzewodowych z dostępem do Internetu. Każdy punkt dostępowy i klienci należący do jednej sieci ESS mają ten sam numer identyfikacyjny nazywany ESSID (od ang. Extended Service Set Identification).



Siec typu ESS.

Źródło: http://pl.wikipedia.org

3. Zastosowane narzędzia

Narzędzia zastosowane do tworzenia pracy to głównie narzędzia za pomocą których zbierane były informacje na temat sieci bezprzewodowych, ale również oprogramowanie, które było wykorzystywane do stworzenia aplikacji.

3. 1. Laptop

Asus Eee PC 1000H jest to 10” netbook z procesorem Intel Atom 1,6GHz oraz 1GB pamięci RAM. Netbook ten idealnie nadaje się do wardrivingu ponieważ jest mały (265,9x191,3x38,1mm) i lekki (1450g). Dzięki zastosowanemu procesorowi Intel Atom pobiera o wiele mniej mocy jak dotychczas powszechnie stosowane procesory w laptopach, dzięki czemu z baterią 6600mAh przy włączonych wszystkich podzespołach (takich jak np. karta Wi-Fi i BlueTooth) wytrzymuje ponad 2 godziny ciągłej pracy.

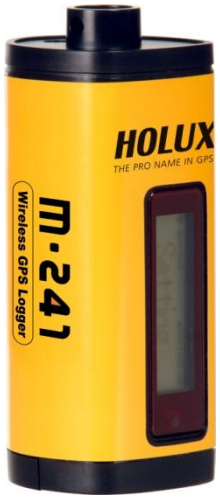
Netbook Asus Eee PC 1000H.

Źródło: http://netbook-review.com

3. 2. Odbiornik GPS

W celu określenia pozycji w której znalazłem sieć potrzebny był mi odbiornik GPS. Wykorzystałem do tego celu bezprzewodowy odbiornik Holux M-241. Jest to bardzo dokładny GPS oparty o chipset MTK, który cechuje niski pobór mocy oraz wysoka dokładność i szybkość z jaką jest w stanie odczytać aktualną pozycję GPS (około 1 sekunda).

Holux M-241 na jednej baterii AA wytrzymuje około 12 godzin, dzięki czemu nie jest potrzebne dodatkowe zasilanie. Odbiornik podłączony był do laptopa za pośrednictwem BlueTooth oraz darmowego oprogramowania GPSd.



Odbiornik GPS Holux M-241.

Źródło: http://www.holux.com

3. 3. Karty Wi-Fi

Do szukania sieci bezprzewodowych wykorzystałem parę podłączonych równolegle kart bezprzewodowych. Główna kartą była zintegrowana w netbooka karta RaLink RT2860, pracująca na chipsecie marki RaLink. Kolejnymi kartami były karty Sagem XG-76NA 802.11bg oparte o chipset ZyDAS ZD1211. Sagem XG-76NA jest kartą WiFi, którą standardowo dostajemy przy zakupie Neostrady z routerem Livebox (Sagem F@st 3202) w Telekomunikacji Polskiej. Jako sterownik dla kart Sagem wykorzystałem compat-wireless-2009-11-03 z poprawkami:

- channel-negative-one-maxim.patch
- mac80211.compat08082009.wl_frag+ack_v1.patch
- zd1211rw-inject+dbi-fix-2.6.26.patch

Karta Wi-Fi Sagem XG-76NA.

Źródło: www.sagem.com

3. 4. System operacyjny

Debian to darmowy system operacyjny z otwartym kodem źródłowym (od ang. open source). System operacyjny to zestaw podstawowych programów i narzędzi, które umożliwiają komputerowi działanie. Debian używa jądra (jest to rdzeń systemu operacyjnego) Linux, ale większość podstawowych narzędzi systemu pochodzi z projektu GNU (uniksopodobny system operacyjny złożony wyłącznie z wolnego oprogramowania). Dlatego system nazywa się Debian GNU/Linux. Debian cieszy się opinią stabilnego systemu o wysokiej jakości oraz łatwego do aktualizacji. Ze względu na dbałość o jakość i bezpieczeństwo dystrybucji, nowe wersje stabilne pojawiają się rzadko. W wersji stabilnej zmiany polegają prawie wyłącznie na naprawianiu problemów dotyczących bezpieczeństwa (przy czym do dystrybucji nie są wprowadzane nowe wersje pakietów, które mogą spowodować nowe problemy - poprawiane są jedynie błędy krytyczne dla bezpieczeństwa systemu). W mojej pracy wykorzystywałem właśnie system Debian GNU/Linux 5.0.7 (lenny), czyli w aktualnej, na dzień pisania pracy wersji stabilnej.

Logo systemu operacyjnego Debian GNU/Linux.

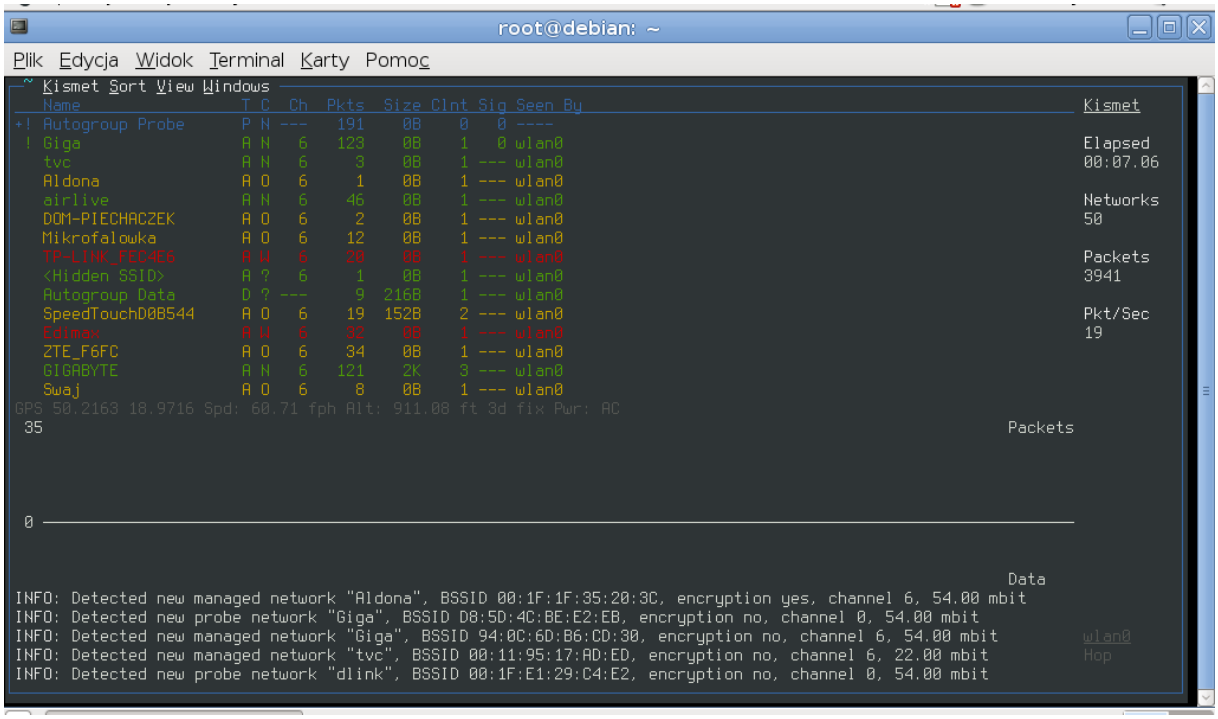
Źródło: www.debian.org

3. 5. Program Kismet

Kismet jest darmowym (na licencji GNU General Public License) programem komputerowym pozwalającym na pasywne wykrywanie bezprzewodowych sieci lokalnych (WLAN), działającym pod kontrolą systemów operacyjnych takich jak Linux, FreeBSD, NetBSD, OpenBSD i Mac OS X. Autorem programu jest Mike "dragorn" Kershaw, a oficjalna strona programu znajduje się pod adresem http://www.kismetwireless.net. Kismet działa praktycznie ze wszystkimi kartami Wi-Fi, które obsługują tryb promiscuous (tryb nasłuchiwania, tryb mieszany, dosł. tryb promiskuitywny, od ang. promiscuous mode). Kismet pozwala na przechwytywanie ramek warstwy drugiej Modelu Osi (warstwa łącza danych) 802.11b, 802.11a oraz 802.11g. Program umożliwia sniffowanie (sniffer jest to najczęściej program komputerowy, którego zadaniem jest przechwytywanie i ewentualnie analizowanie danych przepływających w sieci), oraz posiada pewne cechy systemu IDS (systemy wykrywania i zapobiegania włamaniom, od ang. Intrusion Detection System) dla sieci 802.11.

Kismet w porównaniu do większości programów tego typu wykrywa sieci pasywnie, co oznacza, że nie wysyła żadnych pakietów, które mogłyby by być wykryte. Program potrafi wykrywać zarówno punkty dostępowe (od ang. access point) jak i klientów oraz powiązania między nimi. Częścią programu jest również prosty system IDS, który pozwala na wykrywanie aktywnych prób wyszukiwania sieci bezprzewodowych, które to prowadzi np. program NetStumbler, jak i również wykrywać różnego rodzaju ataki na sieci bezprzewodowe. Kismet potrafi także zapisywać przechwycone pakiety w formacie

programów Wireshark/Tcpdump oraz Airsnort, przez co możliwa jest również późniejsza ich analiza za pomocą powyższych programów. W celu wykrycia jak największej ilości sieci bezprzewodowych program używa techniki "przeskoków kanałów" (od ang. channel hopping), co oznacza, że nie stosuje prostej sekwencji przeskoków między kanałami, jak np. sekwencja 1-2-3-4-5-6-7-8-9-10-11-12-13-14, ale sekwencję, która pozwala przeskakiwać przez kolejne odległe od siebie kanały np. 1-6-11-2-7-12-3-8-13-4-9-14-5-10. Standardowo Kismet przeskakuje między 3 kanałami w ciągu jednej sekundy, przy czym najwięcej czasu spędza na kanałach 1,6,10 oraz 11, ponieważ są to kanały ustawiane standardowo przez producentów bezprzewodowych routerów, m.in. takich popularnych dostawców Internetu jak TP, Netia czy USP. Głównie z uwagi na to, że kanały 1,6 oraz 11 wzajemnie się nie zakłócają. Według badań przeprowadzonych przeze mnie, aż 84% wszystkich znalezionych sieci działa na kanałach 6,11,1,10. Również ze względu na nakładanie się niektórych kanałów na inne, tą metodą można lepiej pokryć pasmo dostępnych częstotliwości i w efekcie przechwycić więcej pakietów, a co za tym idzie wykryć więcej bezprzewodowych sieci lokalnych.



Zrzut ekranu programu Kismet

Źródło: własne

- Kolumna „Name” prezentuje nazwę SSID sieci;

- Kolumna „T” określa typ sieci bezprzewodowej (od ang. Type of WLAN),
  A – punkt dostępowy (od ang. Access Point),
  H – bezprzewodowa sieć o zdecentralizowanej strukturze, w której przyłączone mobilne urządzenia (takie jak np. komputery czy konsole do gier) mogą pełnić funkcje zarówno klienta jak i punktu dostępu (od ang. Ad hoc),
  G – (od ang. Group),
  D – (od ang. Data),
  P – (od ang. Probe).
- Kolumna „C”: szyfrowanie (od ang. Crypt) zastosowane w sieci,
  W – WEP,
  O – (od ang. Other) WPA lub VPN,
  N – brak szyfrowania (od ang. None);
- Kolumna „Ch”: kanał (od ang. Channel) na którym działa sieć;
- Kolumna „Pkts”: ilość przechwyconych pakietów (od ang. Packets);
- Kolumna „Clnt”: ilość klientów sieci (od ang. Clients);
- Kolumna „Sig”: sygnał (od ang. Signal);
- Kolumna „Seen by” wyświetla nazwę interfejsu sieciowego, który znalazł daną sieć;

Na dole głównego okna programu pojawiają się informacje, ostrzeżenia oraz błędy. Przykładowo „new probe network” oznacza przechwycenie pakietów wysyłanych przez oprogramowanie do łączenia się z sieciami bezprzewodowymi, wyświetlana nazwa jest SSID sieci do której wysyłane są pakiety.

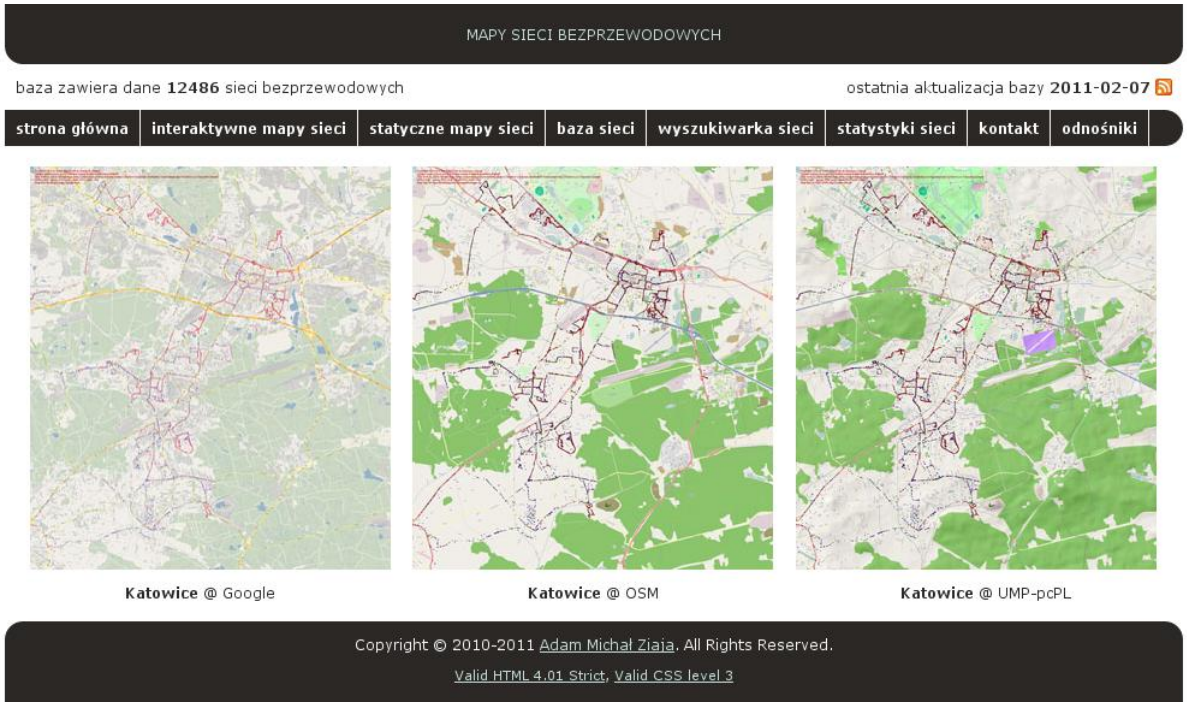
Kismet składa się z trzech oddzielnych części, są nimi:

- drone – zbiera pakiety i wysyła je do serwera programu Kismet;
- serwer – interpretuje pakiety, deszyfruje je, generuje statystyki, zapisuje pakiety i logi, serwer może być używany wraz z wieloma drone, dzięki czemu można stworzyć systemy IDS, IPS (systemy wykrywania i zapobiegania włamaniom, od ang. Intrusion Detection System, Intrusion Prevention System);
- klient – komunikuje się z serwerem Kismet i wyświetla informacje zebrane przez serwer;

Program synchronizował dane z kart Wi-Fi z danymi z GPS, w momencie wykrycia sieci logował również pozycję z odbiornika, dzięki czemu jestem w stanie określić, że daną sieć można wykryć na danej pozycji GPS.

4. Specyfikacja zewnętrzna

Aplikacja jest stroną internetową składającą się zasadniczo z dwóch części, pierwszą jest menu, drugą treść podstrony.



Zrzut ekranu prezentujący aplikację.

Źródło: własne

Strona dzieli się na następujące moduły

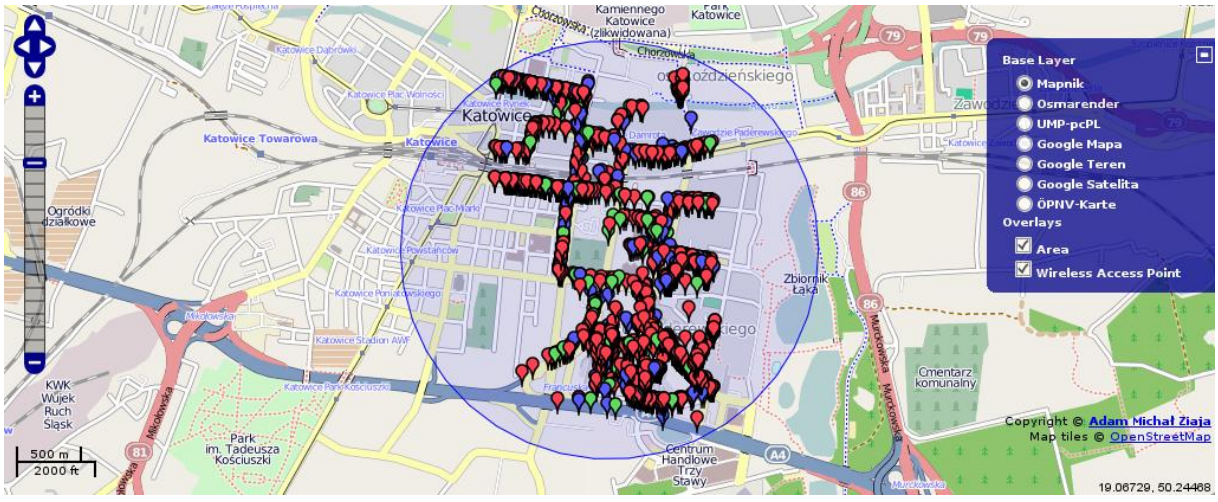
- interaktywne mapy sieci (maps.php);
- statyczne mapy sieci (staticmaps.php);
- baza sieci (db.php);
- sieć bezprzewodowa (ap.php);
- wyszukiwarka sieci (search.php);
- statystyki sieci (stats.php);
- kanał RSS (rss.php);

które zostały szczegółowo opisane poniżej.

4. 1. Moduł: interaktywne mapy sieci (maps.php)

Na podstronie znajdują się interaktywne mapy sieci bezprzewodowych. Siłą napędową map jest darmowa biblioteka JavaScript o nazwie OpenLayers. OpenLayers jest projektem open source i udostępnia bezpłatnie specjalne API podobne do Google Maps czy Bing Maps, dzięki to któremu można budować różnego rodzaju geograficzne aplikacje webowe, takie jak chodź by własne mapy.

Po wejściu na podstronę pojawia się mapa. Na środku pokazują się bezprzewodowe punkty dostępu. Czerwony kolor znacznika oznacza zastosowane szyfrowanie WPA, niebieski WEP, a zielony natomiast brak zastosowanego szyfrowania. Po kliknięciu w znacznik pokazują się bardziej szczegółowe informacje na temat danej sieci. Niebieski obszar w którym ładowane są POI (od ang. point of interest), przy przesunięciu mapy punkty znikają, a następnie ładowane są od nowa, związane jest to głównie z ograniczeniami maszyny Java. Po lewej u góry widnieje panel sterowania mapą, możemy przesuwać mapę za ich pomocą lub też za pomocą myszki, a oddalać oraz przybliżać za pomocą scroll na myszce. W dolnym lewym rogu znajduje się skala mapy, wraz z oddalaniem i przybliżaniem mapy będzie się zmieniać w stosunku do aktualnego zbliżenia. W prawym dolnym rogu natomiast znajduje się aktualna pozycja GPS na której znajduje się kursor myszki. Kawałek wyżej znajdują się informacje na temat praw autorskich do informacji zawartych na mapie jak i warstw mapy. W prawym górnym rogu znajduje się rozwijane menu, z którego możemy wybrać czy mają wyć wyświetlane bezprzewodowe punkty dostępu oraz czy ma być zaznaczony obszar do którego ładowane są punkty na mapie. Kolejną opcją w tym menu jest wybór warstwy mapy, czyli tła pod znacznikami sieci bezprzewodowych, w tym wypadku do wyboru są następujące warstwy:



- Open Street Map Mapnik

Główna warstwa dostarczana jest przez projekt Open Street Map http://openstreetmap.org. Open Street Map jest darmową mapą tworzoną przez wolontariuszy, każdy może edytować tą mapę na takich samych zasadach jak Wikipedię.

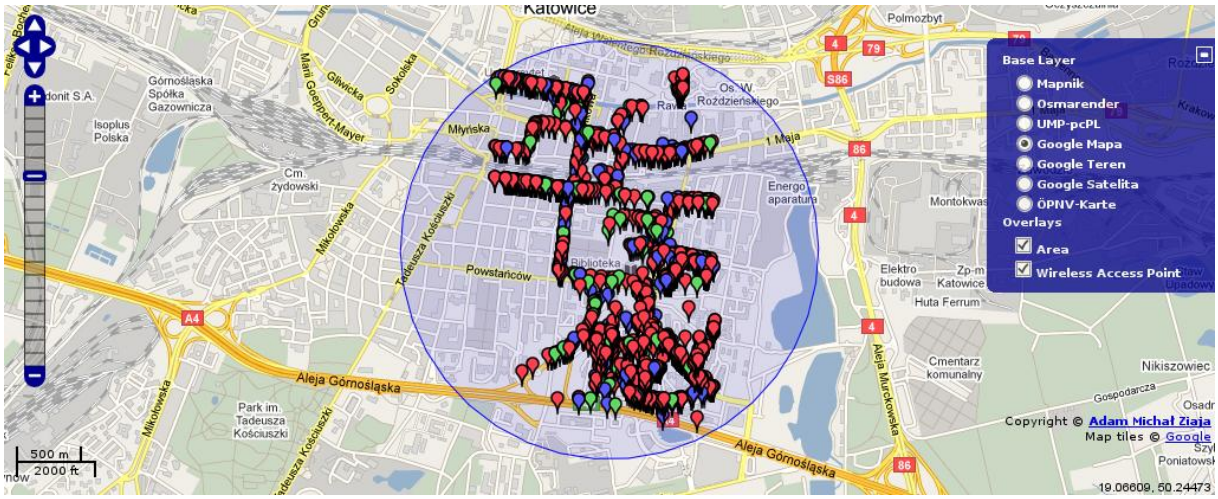
- Open Street Map Osmarender

Kolejna warstwa projektu Open Street Map, została stworzona głównie z myślą o tworzeniu map wektorowych.

- UMP-pcPL

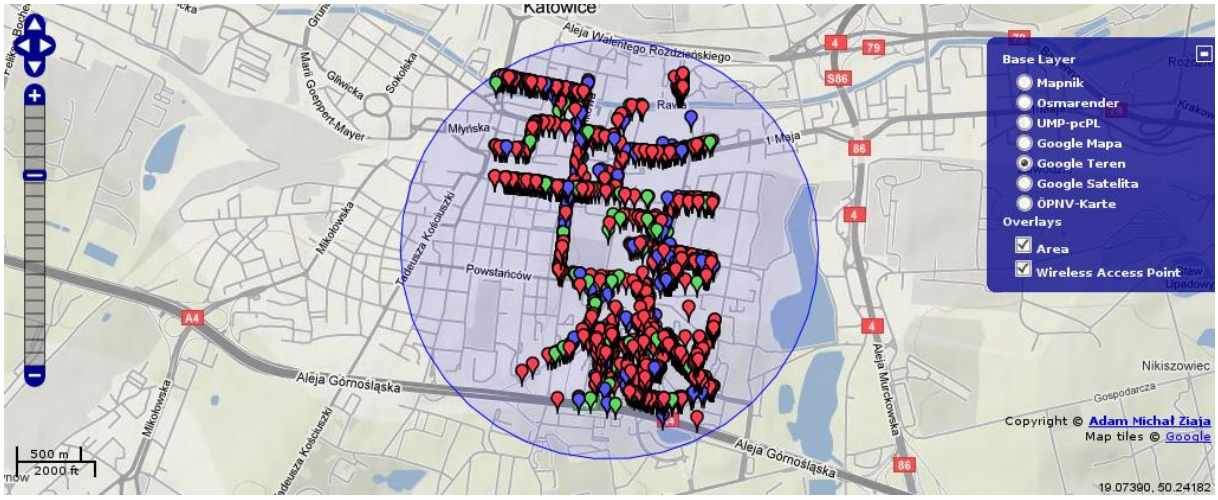
Darmowa mapa polskiego projektu UMP-pcPL http://ump.waw.pl. Projekt jest mapą prawie całej Polski (i stąd skrót "pcPL"). Prawie całej ponieważ nie jest kompletnym i skończonym dziełem, a wciąż rozwijanym tak samo jak Open Street Map również przez

wolontariuszy. Spora część informacji na polskich mapach OSM pochodzi właśnie z projektu UMP-pcPL.



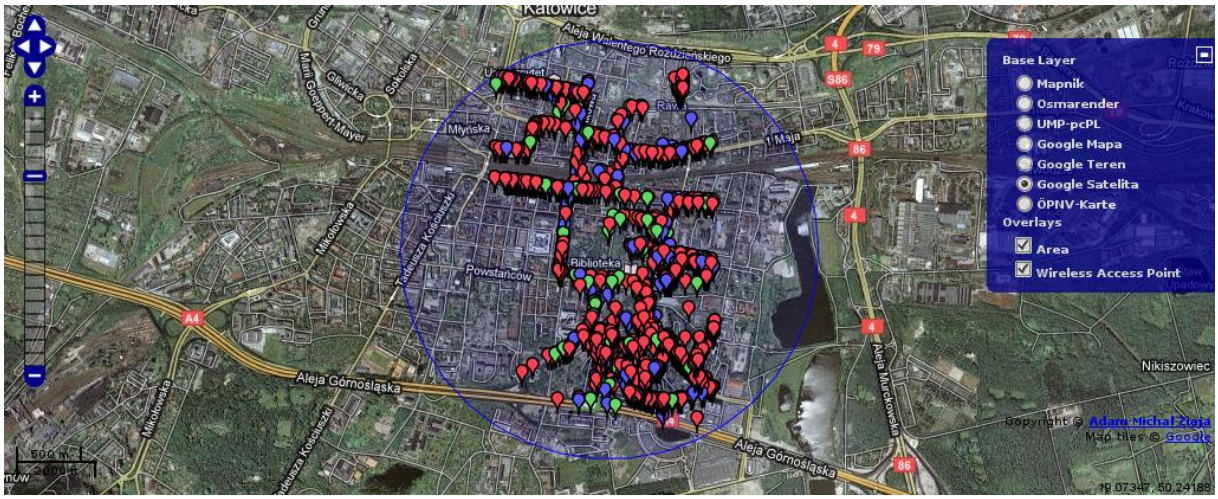
- Google Maps Mapa

Główna warstwa mapy dostarczana przez Google Maps http://maps.google.pl, jest to zwykła mapa.



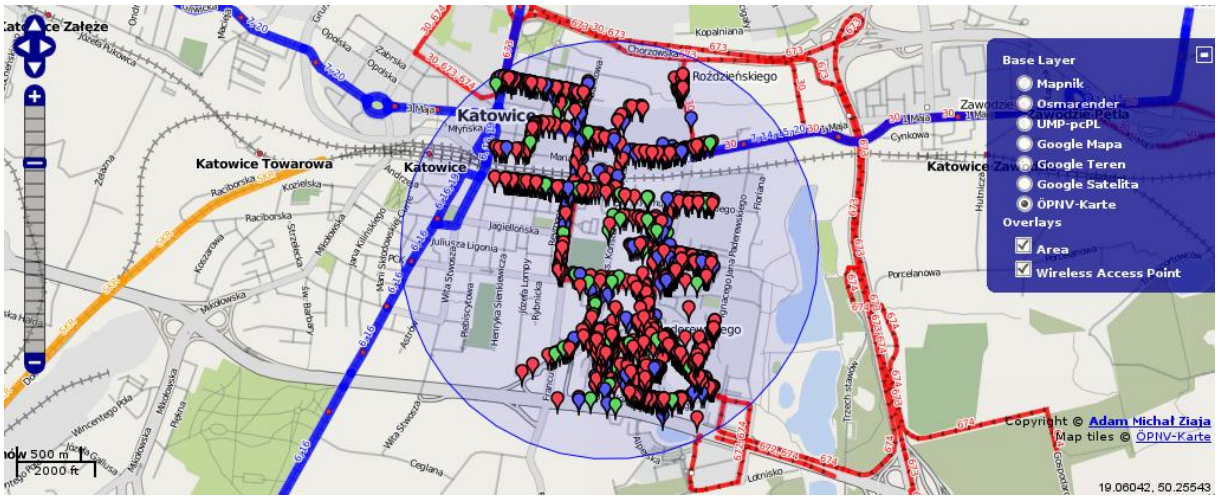
- Google Maps Teren

Terenowa warstwa mapy dostarczana przez Google Maps na której to oznaczone są głównie wypukłości terenu oraz rozlewiska wodne.



- Google Maps Satelita

Mapa stworzona ze zdjęć satelitarnych dostarczana przez Google Maps.



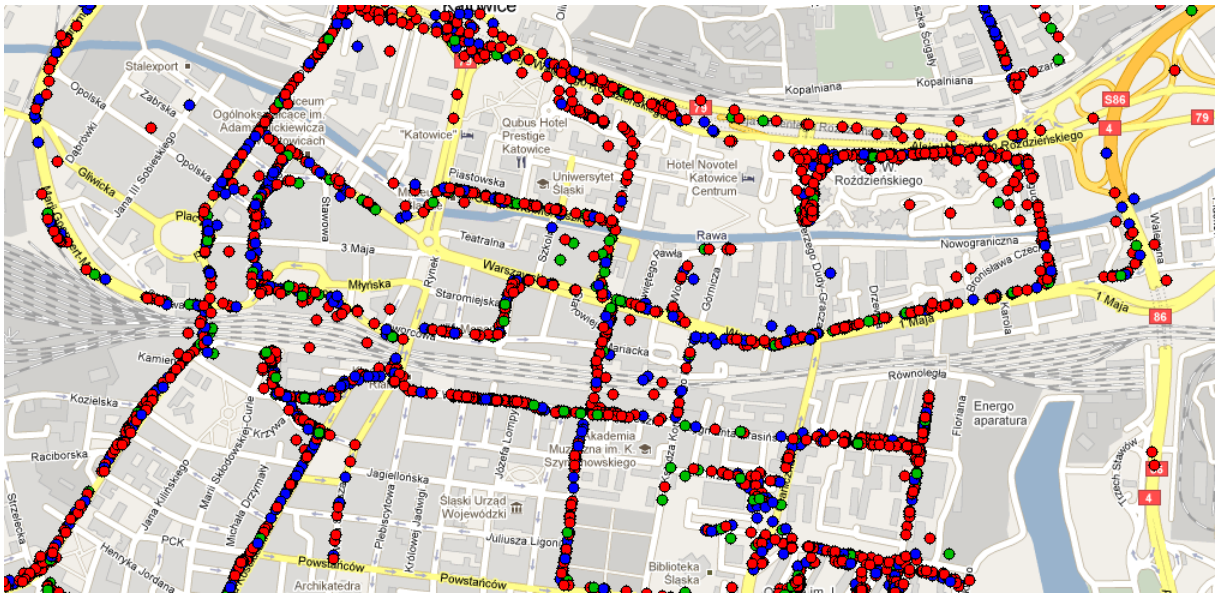
- ÖPNV-Karte

Niemiecka mapa z oznaczoną lokalną komunikacją publiczną dostarczana przez ÖPNV-Karte http://xn—pnvkarte-m4a.de.

4. 2. Moduł: statyczne mapy sieci (staticmaps.php)

Pod tym adresem znajdują się statyczne mapy, czyli nie jest możliwa interakcja z mapą. Mapa generowana jest do grafiki png na podstawie rekordów w bazie i zawiera dokładnie te same punkty, co interaktywne mapy, z tą różnicą, że widać wszystkie punkty naraz, co nie jest możliwe w przypadku interaktywnej mapy ze względu na ograniczenia w Javie. Dzięki takiemu rozwiązaniu łatwo można określić gdzie znajdują się największe skupiska sieci oraz jakie szyfrowanie zostało najczęściej wykorzystane w sieciach w danym

rejonie. Tak samo jak w przypadku map statycznych kolor czerwony oznacza szyfrowanie WPA, kolor niebieski WEP, a zielony brak szyfrowania.



4. 3. Moduł: baza sieci (db.php)

Na tej podstronie znajduje się baza sieci bezprzewodowych przedstawiona w tabelce. Istnieje możliwość sortowania tabelki po przez linki widniejące w nagłówkach tabeli. Pod tabelą natomiast znajdują się linki do kolejnych stron.

BSSID	Nazwa sieci (SSID)	#	Sposób szyfrowania	Pozycja GPS
00:11:95:33:89:66	Bromba	6	WEP	+50° 12′ 15.65″,+18° 58′ 16.76″ A
00:4F:62:1B:1C:BD	czas	5	brak szyfrowania	+50° 12′ 16.82″,+18° 58′ 16.09″ A
00:90:96:00:00:02	wajha	6	WPA+TKIP, WPA+PSK	+50° 12′ 25.05″,+18° 58′ 22.23″ A
00:1D:92:17:5D:33	Pentagram P 6331-6	6	WPA+TKIP, WPA+PSK	+50° 12′ 20.76″,+18° 58′ 18.11″ A
00:26:91:DB:DE:42	Orange_DE40	6	WPA+TKIP, WPA+PSK, WPA+AES-CCM	+50° 12′ 28.8″,+18° 58′ 23.61″ A
00:1B:2F:64:22:02	Swaj	6	WPA+TKIP, WPA+PSK, WPA+AES-CCM	+50° 13′ 46.27″,+18° 56′ 12.37″ A
00:1D:7D:4B:50:B9	GIGABYTE	6	brak szyfrowania	+50° 12′ 30.54″,+18° 58′ 25.37″ A
00:1F:1F:47:AB:02	Dom	7	WEP	+50° 12′ 34.53″,+18° 58′ 27.05″ A
00:26:ED:98:F6:FC	ZTE_F6FC	6	WPA+TKIP, WPA+PSK	+50° 12′ 33.81″,+18° 58′ 26.91″ A
00:1F:1F:4F:09:32	Edimax	6	WEP	+50° 12′ 36.73″,+18° 58′ 26.44″ A
00:24:D2:96:D9:67	SpeedTouchD0B544	6	WPA+PSK, WPA+AES-CCM	+50° 12′ 49.35″,+18° 58′ 18.87″ A
00:26:91:DB:DB:4E	Orange_DB4C	6	WPA+TKIP, WPA+PSK, WPA+AES-CCM	+50° 12′ 35.91″,+18° 58′ 25.99″ A
00:27:19:FE:C4:E6	TP-LINK_FEC4E6	6	WEP	+50° 12′ 39.24″,+18° 58′ 25.83″ A

Przykładowe informacje na temat sieci prezentowane przez skrypt.

Źródło: własne

Tabela przedstawia kolejno BSSID, SSID, kanał, zastosowany sposób szyfrowania oraz pozycję GPS pod którą wykryto daną sieć bezprzewodową. Zielona litera „A” na końcu pozycji GPS oznacza że został określony dokładny adres dla tej pozycji geograficznej, natomiast szary kolor litery oznacza, że jeszcze nie został określony adres i dane zostaną

dopiero zaktualizowane. Po kliknięciu na pozycję GPS otwiera się http://maps.google.pl z zaznaczonym miejscem w którym znaleziono daną sieć, natomiast po kliknięciu na BSSID otwiera się podstrona z bardziej szczegółowymi informacjami na temat danej sieci.

4. 4. Moduł: sieć bezprzewodowa (ap.php)

To właśnie na tej podstronie wyświetlają się najbardziej szczegółowe informacje na temat poszczególnych sieci bezprzewodowych, takie jak SSID, BSSID, producent urządzenia, kanał, prędkość sieci, sposób szyfrowania, sygnał, pozycja GPS oraz adres, a także informacje na temat czasu kiedy pierwszy raz widziano sieć, kiedy ostatni raz widziano, kiedy siec została dodana do bazy oraz kiedy została w niej zaktualizowana. Natomiast na dole wyświetlają się dwie mapy z zaznaczonym miejscem znalezienia sieci, jedna mapa bardziej zbliżona, druga bardziej oddalona.

SSID (nazwa sieci)	Orange_DE40
BSSID (adres MAC urządzenia)	00:26:91:DB:DE:42
Producent urządzenia	SagemCom
Kanał	6 (2,437 GHz)
Prędkość sieci	54 Mb/s
Sposób szyfrowania	WPA+TKIP, WPA+PSK, WPA+AES-CCM
Sygnał	-86 dBm
Pozycja GPS	+50° 12′ 28.8″,+18° 58′ 23.61″ (50.208001,18.973225)
Adres	Kasztanowa 5, Katowice, Polska
Pierwszy raz widziana	2010-10-08 16:51:56
Ostatni raz widziana	2011-01-02 14:55:13
Dodana do bazy	2011-01-01 19:37:16
Aktualizowana w bazie	2011-01-02 15:22:51

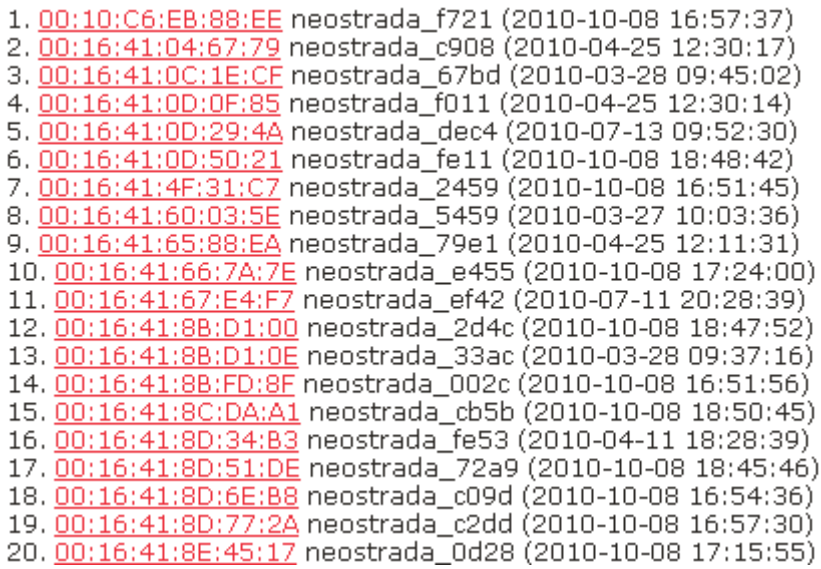
Przykładowe szczegółowe informacje na temat sieci.

Źródło: własne

4. 5. Moduł: wyszukiwarka sieci (search.php)

Pod tym adresem znajduje się wyszukiwarka sieci. Sieci można wyszukiwać po przez podanie BSSID lub SSID. W przypadku BSSID musi zostać podany kompletny BSSID, natomiast w przypadku SSID zostaną odnalezione człony zawierające dany ciąg znaków. Żeby poprawić czytelność wyników liczba rekordów SSID uzależniona jest od ilości wpisanych znaków, dokładnie 10 rekordów na 1 wpisany znak do wyszukiwarki, czyli w

przypadku wpisania „neostrada” limit rekordów będzie wynosić 90 ponieważ wyraz ma 9 liter (znaków).


1. 00:10:C6:EB:88:EE neostrada_f721 (2010-10-08 16:57:37)
2. 00:16:41:04:67:79 neostrada_c908 (2010-04-25 12:30:17)
3. 00:16:41:0C:1E:CF neostrada_67bd (2010-03-28 09:45:02)
4. 00:16:41:0D:0F:85 neostrada_f011 (2010-04-25 12:30:14)
5. 00:16:41:0D:29:4A neostrada_dec4 (2010-07-13 09:52:30)
6. 00:16:41:0D:50:21 neostrada_fe11 (2010-10-08 18:48:42)
7. 00:16:41:4F:31:C7 neostrada_2459 (2010-10-08 16:51:45)
8. 00:16:41:60:03:5E neostrada_5459 (2010-03-27 10:03:36)
9. 00:16:41:65:88:EA neostrada_79e1 (2010-04-25 12:11:31)
10. 00:16:41:66:7A:7E neostrada_e455 (2010-10-08 17:24:00)
11. 00:16:41:67:E4:F7 neostrada_ef42 (2010-07-11 20:28:39)
12. 00:16:41:8B:D1:00 neostrada_2d4c (2010-10-08 18:47:52)
13. 00:16:41:8B:D1:0E neostrada_33ac (2010-03-28 09:37:16)
14. 00:16:41:8B:FD:8F neostrada_002c (2010-10-08 16:51:56)
15. 00:16:41:8C:DA:A1 neostrada_cb5b (2010-10-08 18:50:45)
16. 00:16:41:8D:34:B3 neostrada_fe53 (2010-04-11 18:28:39)
17. 00:16:41:8D:51:DE neostrada_72a9 (2010-10-08 18:45:46)
18. 00:16:41:8D:6E:B8 neostrada_c09d (2010-10-08 16:54:36)
19. 00:16:41:8D:77:2A neostrada_c2dd (2010-10-08 16:57:30)
20. 00:16:41:8E:45:17 neostrada_0d28 (2010-10-08 17:15:55)

Przykładowy wynik wyszukiwania SSID po wpisaniu „neostrada”.

Źródło: własne

4. 6. Moduł: statystyki sieci (stats.php)

Na tej podstronie znajdują się różnego rodzaju statystyki znalezionych sieci bezprzewodowych, związane z regionami administracyjnymi, nazwami sieci, producentami sprzętu, kanałami czy szyfrowaniem. Statystyki szczegółowo zostaną omówione w podsumowaniu.

4. 7. Moduł: kanał RSS (rss.php)

Kanał RSS serwisu, pojawiają się tutaj informacje o aktualizacji bazy danych sieci bezprzewodowych.

5. Specyfikacja wewnętrzna

Tabela do przechowywania informacji na temat sieci bezprzewodowych została utworzona przy pomocy komendy

```
CREATE TABLE wardriving (
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
bssid char(17) NOT NULL UNIQUE,
ssid char(32),
cloaked char(5) NOT NULL,
channel tinyint NOT NULL,
encryption1 text NOT NULL,
encryption2 text default NULL,
encryption3 text default NULL,
maxrate smallint NOT NULL,
manuf text NOT NULL,
gpslat double NOT NULL,
gpslon double NOT NULL,
signal tinyint default NULL,
firstseen datetime NOT NULL,
seen datetime NOT NULL,
adddate datetime NOT NULL,
changedate datetime NOT NULL,
admarea3 text default NULL,
admarea2 text default NULL,
admarea1 text default NULL,
address text default NULL
);
```

Tabela do przechowywania informacji o sparsowanych logach, wykorzystywana głównie do tworzenia kanału RSS została stworzona przy pomocy zapytania

```
CREATE TABLE parser (
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```
logfile char(50) NOT NULL UNIQUE,
adddate datetime NOT NULL
);
```

Tabela do przechowywania logów, w celach statystycznych oraz przechowywania informacji o błędach przy zmiennych pobieranych przez metodę GET została stworzona zapytaniem

```
CREATE TABLE log (
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
ip varchar(15) NOT NULL,
date datetime NOT NULL,
error ENUM('n','y') NOT NULL,
request text NOT NULL,
string text NOT NULL
);
```

Struktura katalogów aplikacji wygląda następująco

`css` – przechowywane są tutaj arkusze stylów;

`img` – obrazki wykorzystane na stronie;

`inc` – includowane pliki;

`maps` – wygenerowane, statyczne mapy;

`adm` – znajduje się tutaj panel administratora;

`adm/cache` – cache, warstwy statycznych map;

`adm/kismet` – logi programu Kismet;

`adm/kismet/todo` – logi do sparsowania;

`adm/kismet/done` – logi sparsowane, przenoszone są tutaj po sparsowaniu przez skrypt adm/mysql.php;

5. 1. Moduł: interaktywne mapy sieci (maps.php)

Skrypt wyświetlający interaktywne mapy sieci bezprzewodowych, w większości napisany jest w języku JavaScript na podstawie dokumentacji OpenLayers API, na którym to

API (od ang. Application Programming Interface) został oparty. Mapy opierają się również na skrypcie JS podchodzącym z darmowego projektu OpenStreetMap.

Mapa przy przesunięciu każdorazowo odpytuje bazę danych, odpowiada za to kod

```
pois.destroy();
pois = new OpenLayers.Layer.Text( "Wireless Access Point",
{location:"./fetch.php?lat=" + center.lat + '&lon=' + center.lon,
projection: map.displayProjection});
map.addLayer(pois);
```

Kod odpytuje skrypt fetch.php przekazując przez GET aktualnie widzianą pozycję, który to skrypt wywołuje następnie zapytanie do bazy MySQL

```
SELECT * FROM wardriving WHERE gpslat BETWEEN $minlat AND $maxlat
AND gpslon BETWEEN $minlon AND $maxlon
```

W odpowiedzi zwracając listę sieci bezprzewodowych wraz z podstawowymi informacjami.

Prezentowane warstwy map definiowane są przy pomocy funkcji map.addLayers() w następujący sposób

```
map.addLayers([layerMapnik,layerOsmarender,layerUMP,layerGoogleStree
ts,layerGoogleTerrain,layerGoogleAerial,layerOPNV]);
```

Każda nazwa odnosi się do wcześniej zdefiniowanej zmiennej, która to wywołuje funkcję OpenLayers.Layer.OSM(), istnieje tutaj możliwość definiowania dowolnych serwerów z warstwami, przykładowo mapa Google została zdefiniowana w następujący sposób

```
var layerGoogleStreets = new OpenLayers.Layer.OSM("Google Mapa",
["http://mt0.google.com/vt/lyrs=m&hl=pl&x=${x}&y=${y}&z=${z}",
"http://mt1.google.com/vt/lyrs=m&hl=pl&x=${x}&y=${y}&z=${z}"],
{numZoomLevels: 20}
);
```

Gdzie fragment kodu

```
http://mt0.google.com/vt/lyrs=m&hl=pl&x=${x}&y=${y}&z=${z}
```

Odpowiada za fragment mapy, a {x}, {y}, {z}, za pozycję XYZ. Każdy fragment mapy w przypadku OpenStreetMap, UMP-pcPL czy Google Maps ma 256 na 256 pikseli oraz określany jest przez pozycję XYZ, w związku z tym wszystkie mapy budowane tym

sposobem możliwe są do wyświetlenia jako kolejna warstwa. Podanie kolejnych serwerów warstw dla danej mapy w znacznym stopniu przyspiesza jej ładowanie. Nie każdą mapę da się zdefiniować w ten sposób, przykładowo Zumi definiuje warstwy własnym sposobem, najprawdopodobniej w celu uniemożliwienia zdefiniowania ich w innym skrypcie, warstwa tutaj ma rozmiar 250 na 250 pikseli i jej adres wygląda następująco

```
http://mimgl4.onet.pl/001000t01s09x00052y00019.png?7c2b58
```

5. 2. Moduł: baza sieci (db.php)

Skrypt ten wyświetla w tabeli informacje o sieciach bezprzewodowych na podstawie zapytań do bazy danych MySQL, stąd też nazwa „db", od database. Pobierane są trzy zmienne za pomocą GET, a są nimi

- page - odpowiada za numer strony z wynikami;
- sort - odpowiada za sortowanie wyników w zależności od BSSID, SSID, sposobu szyfrowania oraz pozycji GPS;
- order - odpowiada za malejące i rosnące sortowanie;

W tabeli wyświetlane są podstawowe dane o znalezionych sieciach bezprzewodowych, które znajdują się w bazie, czyli

- BSSID

Wyświetlane są dane z bazy danych z kolumny „bssid", dane przechowywane są w zmiennej $bssid.

- SSID

Wyświetlane są dane z kolumny „ssid", dane przechowywane są w zmiennej $ssid

```
$ssid=htmlspecialchars(mysql_result($result,$i,"ssid"),ENT_QUOTES);
```

W tym przypadku dzięki zastosowaniu funkcji htmlspecialchars() znaki specjalne, które mogłyby być interpretowane jako HTML zostaną zmienione na string, w związku z tym, np. znak

- '&' zostanie zamieniony na '&'
- '"' zostanie zamieniony na '"' (jeśli ENT_NOQUOTES nie jest ustawiony)
- ''' zostanie zamieniony na ''' (tylko jeśli ENT_QUOTES jest ustawiony)

- '<' zostanie zamieniony na '<'
- '>' zostanie zamieniony na '>'

w przeciwnym wypadku, gdyby nie została zastosowana funkcja htmlspecialchars() (lub inna podobna jak np. addslashes()) w przypadku wyników np.

- KoNiK's Realm
- Marek's Home
- Kanar's Wifi
- Poul Jozefiak's Network

wystąpiły by problemy ze względu na znak specjalny „'", który zostałby zinterpretowany jako część kodu, a nie string.

W przypadku SSID bazie sprawdzana jest również kolumna „cloaked", która przyjmuje wartości true oraz false w zależności od tego czy nazwa sieci (SSID) jest ukryta, w przypadku wyniku true dla danej sieci zmienna przyjmuje wartość

```
$cloaked="<span class=\"szary\">nazwa sieci została ukryta</span>";
```

Dzięki czemu zamiast SSID w tabeli zostanie wyświetlona informacja w kolorze szarym, że dana siec ma wyłączone rozgłaszanie nazwy;

- Kanał

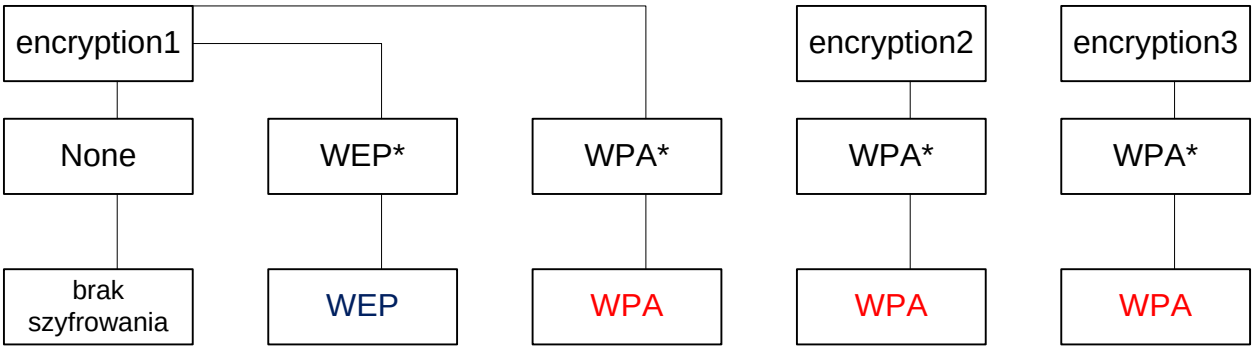
Wyświetlane są dane z kolumny „channel", w przypadku gdy kanał nie został wykryty w bazie przyjmuje wartość 0, w tym wypadku zmienna przyjmuje wartość

```
$channel="<span class=\"szary\">?</span>"
```

Dzięki czemu zamiast „0" wyświetlany jest szary znak zapytania;

- Sposób szyfrowania

Wyświetlane są dane z kolumn „encryption1", „encryption2" oraz „encryption3". W przypadku kolumny „encryption1" skrypt sprawdza czy przyjmuje wartość „None" jeśli tak to oznacza to, że sieć nie jest szyfrowana i zmienna przyjmuje wartość;



Schemat działania skryptu.

Źródło: własne

- Pozycja GPS

Wyświetlana jest pozycja GPS w formacie DMS dzięki zastosowaniu funkcji DECtoDMS() znajdującej się w pliku inc/gps.php. Pozycja GPS w bazie danych zapisana jest w formacie DEC, z tego również formatu tworzony jest link do mapy. W tej kolumnie prezentowanych danych sprawdzana jest również kolumna address z bazy danych, jeśli znajdują się w niej informacje na temat adresu obok pozycji GPS wyświetlana jest zielona litera „A", a przeciwnym wypadku wyświetlana jest szara litera „A".

5. 3. Moduł: sieć bezprzewodowa (ap.php)

Skrypt ten jest niejako integralną częścią db.php, a nazwa „ap" pochodzi od Access Point. Podstrona ta wyświetla najbardziej szczegółowe informacje jakie są dostępne w bazie na temat danej sieci bezprzewodowej. Pobierana jest jedna wartość, mianowicie BSSID, do zmiennej $bssid

```
$bssid=mysql_real_escape_string(htmlspecialchars(addslashes($_GET['b
ssid']),ENT_QUOTES));
```

Przy pobieraniu danych do zmiennej zastosowane zostały, aż trzy zabezpieczenia przeciw (My)SQL Injection. SQL Injection (z ang. dosłownie zastrzyk SQL) jest to luka w zabezpieczeniach aplikacji internetowych, polegająca na nieodpowiednim filtrowaniu lub niedostatecznym typowaniu i późniejszym wykonaniu danych przesyłanych w postaci zapytań SQL do bazy danych. Podatne są na nią systemy złożone z warstwy programistycznej (przykładowo skrypt w PHP, ASP, JSP itp.) dynamicznie generującej zapytania do bazy danych (MySQL, PostgreSQL itp.). Wynika on zwykle z braku doświadczenia lub wyobraźni programisty. Funkcja htmlspecialchars() została już przeze mnie omówiona wcześniej,

funkcja addslashes() zwraca string i poprzedza znakiem „\" wszystkie znaki specjalne, które mogłyby być zinterpretowane jako część kodu, natomiast funkcja mysql_real_escape_string() została stworzona specjalnie do filtrowania w celu zapobiegania atakom typu MySQL Injection.

Cześć zmiennych jest dokładnie tak samo pobierana jak w przypadku db.php, w przypadku zmiennej $channel, która prezentuje kanał na którym działa siec bezprzewodowa dodatkowo zostały dopisane informacje na temat częstotliwości na której działa sieć

```
if ($channel==1) {$frequency=" (2,412 GHz)";}

if ($channel==14) {$frequency=" <span class=\"czerwony\">(2,484 GHz)
W Polsce tylko częstotliwości od 2,4000 do <b>2,4835 GHz</b> nie
wymagają koncesji!</span>";}
```

Kanał 14 został specjalnie oznaczony z uwagi, że w Polsce stosowanie tego kanału bez koncesji jest nielegalne.

W skrypcie zostało dodane również logowanie, które w późniejszym czasie może być wykorzystane do stworzenia statystyk wyszukiwanych sieci, jak również w celu poznania metod ataków SQL Injection na skrypt, tak więc w przypadku braku błędów, czyli gdy wynik z bazy danych będzie równy jednemu wersowi zostaje wykonane zapytanie

```
$query="INSERT INTO log (ip,date,error,request,string)
VALUES('".$_SERVER['REMOTE_ADDR']."','".date('Y-m-d
H:i:s')."','n','".addslashes($_SERVER['REQUEST_URI'])."','".addslash
es($_GET['bssid'])."')";
```

Zapytanie to loguje adres IP, datę, adres URL oraz wartość zmiennej $bssid. Natomiast w przypadku błędu, czyli ilości wersów równej 0 działanie zostaje odnotowane w logach po przez zapytanie, które zapisuje dokładnie te same dane co zapytanie w przypadku braku błędów, z tym wyjątkiem, że w kolumnie „error" zamiast wartości „n" (od ang. no), zostaje wpisana wartość „y" (od ang. yes), która oznacza, że wystąpił błąd. Praktycznie zawsze oznacza to próbę ataku, ponieważ ap.php linkowane jest jedynie w miejscach w których siec występuje w bazie danych.

5. 4. Moduł: wyszukiwarka sieci (search.php)

Nazwa podstrony pochodzi od ang. „szukaj", tak więc podstrona służy do wyszukiwania rekordów w bazie danych. Po wejściu na podstronę prezentowany jest formularz pozwalający na wyszukiwanie sieci po przez BSSID oraz SSID. Zmienne zabezpieczone są tak samo jak w przypadku db.php oraz ap.php za pomocą funkcji

mysql_real_escape_string(), htmlspecialchars() oraz addslashes(). W celu uniknięcia nadużyć związanych z wyszukiwarką takich jak przykładowo wpisanie jednego znaku, a również w celu uniknięcia zbytecznych ograniczeń wyszukiwarki takich jak przykładowo wyszukiwanie dokładnie podanego stringa zastosowana została funkcja strlen(). Funkcja ta przyjmuje na wejście string, a zwraca liczbę znaków występująca w stringu. Liczba znaków zostaje pomnożona razy 10 i przypisana do zmiennej w przypadku BSSID $bssidlimit, w przypadku SSID $ssidlimit, a następnie wykorzystana jest do limitowania wyników zapytania. W przypadku jeśli $bssidlimit lub $ssidlimit przyjmuje wartość większą od 100 to zostaje nadpisana wartością 100, dzięki czemu odgórnym limitem wyników jest 100. W celu lepszego zobrazowania, jeśli np. do wyszukiwarki SSID wpiszemy słowo „neostrada", to zawiera ono 9 znaków, tak więc 9 zostaje pomnożone przez 10 i $ssidlimit przyjmuje wartość 90, tym samym ograniczając liczbę wyników do 90.

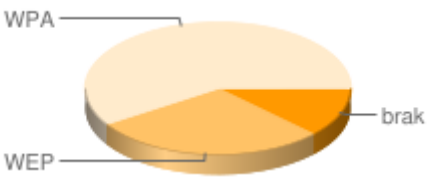
W tym wypadku również tak samo jak w przypadku ap.php zostało dodane logowanie na tej samej zasadzie.

5. 5. Moduł: statystyki sieci (stats.php)

Podstrona ta służy prezentowaniu statystyk na temat zebranych informacji o sieciach bezprzewodowych. Nazwa „stats" oznacza z ang. statystyki. Statystyki generowane są na podstawie zapytań do bazy danych. Dane z geocodowania przechowywane są w kolumnach admarea3 - miasto, admarea2 - powiat, admarea1 – województwo. W celu wygenerowania statystyk miast zostało zastosowane zapytanie

```
SELECT admarea3 FROM wardriving GROUP BY admarea3 ORDER BY
COUNT(admarea3) DESC;
```

Zapytanie to wyciąga z bazy nazwy miast, następnie grupuje je przez nazwę miasta, a na koniec sortuje malejąco wg ilości występowania. Została tutaj również zastosowana funkcja round(), która dostaje na wejście liczbę, która ma zostać zaokrąglona, oraz jako drugi parametr liczbę odpowiadająca za ilość cyfr po przecinku. Praktycznie wszystkie statystyki tworzone są tą samą zasadą. Wykresy generowane są dynamicznie po przez Google Chart API, które szczegółowo zostało opisane pod adresem http://chart.apis.google.com.



Przykładowy wykres wygenerowany po przez Google Chart API.

Źródło: własne

5. 6. Moduł: kontakt (contact.php)

Prosty formularz kontaktowy, wykorzystujący funkcję mail(). Dodatkowo został podany e-mail ukryty po przez reCAPTCHA, w celu uniknięcia dodania adresu e-mail do bazy spamerów.

5. 7. Moduł: sygnatura (sig.php)

Skrypt generujący sygnaturkę w formacie png. Dzięki wpisom w .htaccess

```
RewriteEngine On
RewriteRule sig.png sig.php
```

plik php przez serwer odczytywany jest jako php, natomiast przez klienta interpretowany jako png. Na samym początku skryptu została zastosowana funkcja header() w celu modyfikacji nagłówków pliku i interpretacji go jako obrazka po przez przeglądarkę klienta

```
header('Content-type: image/png');
```

Następnie generowany jest obrazek

```
$obrazek=ImageCreate(234,60)
```

Wymiary obrazka to 234 na 60 pikseli, co odpowiada standardowi pół bannera (od ang. half banner), następnie definiowane są czcionki

```
$czcionka_verdana='Verdana.ttf';
```

Bardzo istotnym elementem jest tutaj wielkość liter, ponieważ system Linux rozróżnia duże/małe litery w przeciwieństwie do systemu Windows. Następnie definiowane są kolory

```
$kolor_czerwony=ImageColorAllocate($obrazek,255,0,0);
$kolor_zielony=ImageColorAllocate($obrazek,0,190,0);
$kolor_niebieski=ImageColorAllocate($obrazek,0,0,255);
```

Funkcja imagecolorallocate() przyjmuje na wejście cztery parametry, pierwszy to obrazek, kolejne trzy to kolory RGB (od ang. Red-Green-Blue) czyli czerwony, zielony oraz niebieski. Kolory przyjmują wartość od 0 do 255. Następnie nanoszony jest tekst za pomocą funkcji ImageTTFText()

```
ImageTTFText($obrazek,9,0,6,16,$kolor_szary,$czcionka_georgia,"WARDR
IVING & WARCHALKING");

ImageTTFText($obrazek,9,0,5,15,$kolor_czarny,$czcionka_georgia,"WARD
RIVING & WARCHALKING");
```

Każdy tekst został dwa razy powielony, ponieważ pierwszy tworzy cień w kolorze szarym, a następnie nanoszony jest tekst w innym kolorze, przesunięty o 1 piksel.

```
ImagePng($obrazek);

ImageDestroy($obrazek);
```

Na koniec tworzony jest obrazek png, a następnie „niszczony", w celu wygenerowania go ponownie przy odświeżeniu.

5. 8. Moduł: robots (robots.txt)

Plik zawiera informacje dla wyszukiwarek internetowych, a konkretniej dla ich robotów, które zbierają informacje na stronach internetowych.

```
User-agent: *

Allow: /

Disallow: /adm/
```

Powyższy zapis oznacza, że każdy robot może indeksować wszystko poza katalogiem adm, w którym to przechowywane są skrypty administracyjne.

Każdy skrypt znajdujący się w folderze adm wymaga podania hasła, obsługiwane jest to przez skrypt access.php, oparty jest o sesje w PHP.

```
session_start();

session_register("xEpEcr5baWar");
```

Skrypt rejestruje sesję (od PHP 5.3 rejestracja nie jest wymagana), następnie sprawdza czy istnieje

```
if(empty($_SESSION["xEpEcr5baWar"])){$_SESSION["xEpEcr5baWar"]=0;}
```

Jeśli nie, przypisuje jej wartość zero

```
if($_SESSION["xEpEcr5baWar"]!=1338){

   if(!empty($_POST["password"])){

if($_POST["password"]=="tr6vECaD5sTA"){$_SESSION["xEpEcr5baWar"]=133
8;} else {ShowLogin("<p class=\"czerwony\">Złe hasło!</p>");die;}
```

```
    } else {ShowLogin();die;}
}
```

Skrypt wyświetla formularz do podania hasła, w przypadku jeśli hasło zostanie wpisane poprawnie użytkownik zostaje zalogowany i może zobaczyć dalszą część danego skrypty, w przeciwnym wypadku zostaje wywołana funkcja die(), która równoważna jest z funkcją exit(), natychmiastowo kończy ona wykonywanie skryptu, przez co dalsza część kodu nie jest wykonywana.

5. 9. Moduł: parsowanie logów (adm/mysql.php)

Najważniejszy skrypt, odpowiadający za parsowanie logów programu Kismet w formacie XML przy pomocy funkcji SimpleXML(). Mimo że rozszerzenie pliku to netxml, a nie xml to jest to zwykły plik XML. Na samym początku skrypt przy pomocy funkcji glob() pobiera wszystkie nazwy pasujące do zdefiniowanej ścieżki

```
$files = glob('./kismet/todo/Kismet-*.netxml');
```

Gwiazdka w tym zapisie odpowiada za dowolny ciąg znaków. Następnie wywoływana jest pętla, która to wywołuje skrypt dla każdego niesparsowanego logu znajdującego się w folderze adm/kismet/todo/ (do zrobienia, od ang. to-do)

```
foreach ($files as $kismetlog) {
```

Jeśli plik istnieje

```
if (file_exists($kismetlog)) {
$xml=simplexml_load_file($kismetlog);
```

Wywoływana jest funkcja simplexml_load_file(), która za parametr przyjmuje plik XML, w tym wypadku log programu Kismet. Następnie dla każdej infrastrukturalnej sieci, która posiada w logu pozycję GPS (istnieje możliwość, że GPS nie złapał pozycji przed znalezieniem sieci, może się tak dziać chociażby przy wyjazdach z tunneli, przykładowo pod rondem w Katowicach)

```
foreach ($xml->{'wireless-network'} as $wirelessnetwork){
if($wirelessnetwork['type']=='infrastructure'){
if($wirelessnetwork->{'gps-info'}->{'avg-lat'}!=0&&$wirelessnetwork-
>{'gps-info'}->{'avg-lon'}!=0&&$wirelessnetwork->SSID-
>encryption[0]!=''){
```

Skrypt odpytuje bazę danych zapytaniem

```
$query="SELECT * FROM wardriving WHERE bssid = '".$wirelessnetwork-
>BSSID."'";
```

Sprawdza czy dana sieć już została znaleziona, bazując na unikalnym (w teorii) adresie BSSID. Jeśli wynik zapytania zwrócił 0 rekordów skrypt dodaje nową sieć do bazy danych. Na samym początku sprawdzany jest adres MAC

```
$mac=$wirelessnetwork->BSSID;
list($mac1, $mac2, $mac3)=explode(":", $mac);
$mac24="$mac1:$mac2:$mac3";
$query="SELECT * FROM manuf WHERE mac='".$mac24."'";
$result=mysql_query($query);
$num=mysql_num_rows($result);
if($num>0) {
      $manuf=mysql_result($result,$i,"manuf");
}
```

Przy pomocy funkcji explode() rozbijany jest on pomiędzy znakiem dwukropka w celu pobrania tylko pierwszych 24 bitów, które odpowiadają za producenta karty sieciowej. Tworzone jest zapytanie do tabeli manuf, jeśli adres MAC występuje w tabeli nazwa producenta zostaje przypisana do zmiennej $manuf.

Postanowiłem nie korzystać z danych na temat producenta karty sieciowej zawartych w logach programu Kismet, ponieważ często są one nieaktualne na chwilę znalezienia sieci, w tym wypadku musiałbym co jakiś czas przeszukiwać całą bazę i uzupełniać rekordy. Z tego powodu wykorzystałem bazę darmowego programu Wireshark, służącego do analizowania ruchu sieciowego. Jego baza danych również jest darmowa i mieści się pod adresem http://anonsvn.wireshark.org/wireshark/trunk/manuf. W tym celu napisałem prosty skrypt pod system Linux, który to tworzy skrypt SQL dzięki, któremu można utworzyć tabelę w bazie danych zawierającą dane z powyższej bazy programu Wireshark.

```
wget http://anonsvn.wireshark.org/wireshark/trunk/manuf; cat manuf |
grep -v "^#" | grep -v "/" | grep ":" | sed -e 's/[[:space:]]/ /' |
sed -e 's/"//g' | awk '{print "INSERT INTO manuf (mac,manuf)
VALUES(\"" $1 "\",\"" $2 "\");"}' > manuf.sql
```

Po weryfikacji adresu MAC skrypt tworzy zapytanie do bazy jednocześnie parsując logi XML. Jak już wspomniałem parsowanie odbywa się przy pomocy funkcji

simplexml_load_file(), wchodzącą w skład SimpleXML. Strukturę pliku XML można zobaczyć za pomocą pierwszego skryptu podanego w dokumentacji funkcji, czyli

```
$xml = simplexml_load_file('test.xml');
print_r($xml);
```

W celu poprawienia czytelności można dodać tag HTML pre. Istotna część bardzo długiej struktury pliku programu Kismet w interesujących fragmentach wygląda następująco

```
[type] => infrastructure
```

Aby wywołać ten fragment logu trzeba zastosować kod

```
simplexml_load_file('plik.xml')->{'wireless-
network'}['type']=='infrastructure'
```

Analogicznie można wyciągnąć każdy tag z pliku XML

[first-time] => Sun Jan 2 14:51:20 2011	siec pierwszy raz widziana przez jakąkolwiek kartę Wi-Fi
[max-rate] => 54.000000	prędkość sieci
[encryption] => Array [0] => WPA+TKIP [1] => WPA+PSK	zastosowane rodzaje szyfrowania
[essid] => dlink	nazwa sieci Wi-Fi
[BSSID] => 00:19:5B:9B:86:A6	zazwyczaj adres MAC urządzenia (np. routera)
[manuf] => D-Link	producent sprzętu
[channel] => 6	kanał na którym działa sieć
[freqmhz] => 2447 1	częstotliwość działania sieci
[avg-lat] => 50.223862 [avg-lon] => 18.969692 [avg-alt] => 267.000000	pozycja GPS
[wireless-client] => SimpleXMLElement Object	znaleziony klient podłączony do danej sieci bezprzewodowej
[client-mac] => 00:19:5B:9B:86:A6	adres MAC klienta
[client-manuf] => D-Link	producent bezprzewodowej karty sieciowej

Analogicznie w pliku XML wygląda każda siec.

Natomiast jeśli sieć wystąpiła w bazie to skrypt aktualizuje dane. Przykładowo jeśli sieć została ponownie wykryta i posiadała lepszy sygnał

```
$signal=dbquery("SELECT signal FROM wardriving WHERE
bssid='".$wirelessnetwork->BSSID."'","signal");

if($signal='0'||$signal<$wirelessnetwork->{'snr-info'}-
>{'max_signal_dbm'}){

mysql_query("UPDATE wardriving SET gpslat='".$wirelessnetwork-
>{'gps-info'}->{'avg-lat'}."',gpslon='".$wirelessnetwork->{'gps-
info'}->{'avg-lon'}."',signal='".$wirelessnetwork->{'snr-info'}-
>{'max_signal_dbm'}."',changedate='".date('Y-m-d
H:i:s')."',admarea3=NULL,admarea2=NULL,admarea1=NULL,address=NULL
WHERE bssid='".$wirelessnetwork->BSSID."';");
```

Zostaje nadpisana m.in. pozycja GPS, w celu prezentowania jak najdokładniejszych danych.

5. 10. Moduł: geokodowanie adresów (adm/geocode.php)

Jest to innowacyjny mojego autorstwa pomysł wykorzystania danych z Google do tworzenia możliwie najdokładniejszych statystyk bazujących na zgeocodowanych adresach GPS sieci bezprzewodowych. Skrypt ten nie jest częścią mysql.php, ponieważ geocodowanie jest bardzo czasochłonne. Na samym początku skrypt sprawdza, które sieci nie posiadają przypisanego adresu

```
$query="SELECT * FROM wardriving WHERE admarea1 IS NULL OR admarea2
IS NULL OR address IS NULL ORDER BY seen DESC";
```

Google posiada dość restrykcyjne limity na pobieranie danych tego typu, tak więc raz dane są pobierane z publicznego adresu IP pod którym działa skrypt, na przemian z wywołaniem ściągania za pośrednictwem sieci TOR. TOR (od ang. The Onion Router) jest wirtualną siecią komputerową implementującą trasowanie cebulowe drugiej generacji, zapobiegającą analizie ruchu sieciowego i w konsekwencji zapewniającą użytkownikom prawie anonimowy dostęp do zasobów Internetu. Kod odpowiedzialny za to wygląda następująco

```
if ($torify % 2 == 0) {

shell_exec("torify wget -q -O /tmp/".$gpslat.",".$gpslon.".xml
'http://maps.googleapis.com/maps/api/geocode/xml?latlng=".$gpslat.",
".$gpslon."&sensor=false&region=pl&language=pl'");

$xmltmp=file_get_contents("/tmp/".$gpslat.",".$gpslon.".xml");

   sleep(3);

} else {
```

```
$xmltmp=file_get_contents("http://maps.googleapis.com/maps/api/geoco
de/xml?latlng=".$gpslat.",".$gpslon."&sensor=false&region=pl&languag
e=pl");
```

```
   sleep(3);
```

Po każdym pobraniu danych w formacie XML skrypt odczekuje 3 sekundy, ze względu na wspomniane już odgórne limity pobierania danych z Google. Jeśli czasowy limit został przekroczony plik XML zwraca

```
OVER_QUERY_LIMIT
```

W tym wypadku, skrypt odczekuje dodatkowe 10 sekund

```
if($xml->status==OVER_QUERY_LIMIT){ sleep(10); }
```

W przypadku otrzymania statusu OK. wykonuje się normalnie, parsując otrzymany plik XML

```
foreach ($xml->result[0]->address_component as $obj) if($obj-
>type[0]==administrative_area_level_1) {$admarea1=$obj->long_name;}
```

```
$address=$xml->result[0]->formatted_address;
```

Następnie dodaje do bazy otrzymane dane

```
mysql_query("UPDATE wardriving SET
admarea3='".addslashes($admarea3)."',"admarea2='".addslashes($adma
rea2)."',"admarea1='".addslashes($admarea1)."',"address='".addsl
ashes($address)."' WHERE gpslat='".$gpslat."' AND
gpslon='".$gpslon."';");
```

5. 11. Moduł: statyczne mapy sieci (adm/staticmaps.php)

Skrypt służy do generowania statycznych map, wzory pochodzą z dokumentacji projektów OpenLayers oraz OpenStreetMap, w szczególności http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames. Na początku działania skrypt pobiera z bazy danych informacje na temat skrajnych pozycji zgeocodowanych sieci znalezionych w Katowicach

```
$lat=dbquery("SELECT gpslat FROM wardriving WHERE admarea2 =
'Katowice' ORDER BY gpslat DESC LIMIT 1","gpslat");
```

```
$lon=dbquery("SELECT gpslon FROM wardriving WHERE admarea2 =
'Katowice' ORDER BY gpslon DESC LIMIT 1","gpslon");
```

```
$lat=dbquery("SELECT gpslat FROM wardriving WHERE admarea2 =
'Katowice' ORDER BY gpslat ASC LIMIT 1","gpslat");
```

```
$lon=dbquery("SELECT gpslon FROM wardriving WHERE admarea2 =
'Katowice' ORDER BY gpslon ASC LIMIT 1","gpslon");
```

Funkcja map() odpowiedzialna jest za tworzenie mapy, przyjmuje ona parametry

```
map($name, $xmin, $xmax, $ymin, $ymax, $zoom, $circle, $tiles)
```

Nazwa mapy, wymiary w osiach X oraz Y, przybliżenie, wielkość znacznika oraz na jakich warstwach ma być budowana mapa. W trakcie wykonywania skrypt odwołuje się do funkcji map_gettile() w celu pobrania warstwy. Funkcja ta sprawdza czy warstwa jest starsza niżeli ilość dni zdefiniowana na początku skrypty przy pomocy MAP_TILES_CACHE_TIME

```
if(is_file($path)&&time()-
filemtime($path)<MAP_TILES_CACHE_TIME*60*60*24)
```

Następnie, jeśli warstwa mapy jest starsza lub nie istnieje to jest ona pobierana przy pomocy funkcji shell_exec(), w następujący sposób

```
shell_exec("wget -O$path '$url'");
```

Gdzie zmienna $path odpowiada za nazwę pliku i została zdefiniowana wcześniej

```
$path=MAP_TILES_CACHE.$zoom."-".$x."-".$y.".png";
```

Zmienna $url natomiast za adres odnośnika pod którym znajduje się dana potrzebna warstwa mapy, przykładowo w przypadku warstwy Google Maps, skrypt przy pobieraniu warstw mapy wykonuje następujące przykładowe komendy

```
wget -O./cache/google/15-18111-11079.png
http://mt0.google.com/vt/lyrs=m&hl=pl&x=18111&y=11079&z=15

wget -O./cache/google/15-18112-11079.png
http://mt0.google.com/vt/lyrs=m&hl=pl&x=18112&y=11079&z=15
```

Jeśli nie było błędów to tworzy się mapa, następnie nanoszone są sieci bezprzewodowe z bazy danych za pomocą zapytania

```
mysql_query("SELECT gpslat,gpslon,encryption1 FROM wardriving WHERE
gpslon BETWEEN $lonmin AND $lonmax AND gpslat BETWEEN $latmin AND
$latmax AND gpslat!=0 AND gpslon!=0");
```

Rysowane jest koło oznaczające jedną sieć, kolor zależny jest od rodzaju szyfrowania

```
imagefilledarc($map, $xpos, $ypos, $circle-1, $circle-1, 0, 360,
$farba, IMG_ARC_PIE);

imagefilledarc($map, $xpos, $ypos, $circle, $circle, 0, 360,
$kolor_czarny, IMG_ARC_NOFILL);
```

Pierwsza linia oznacza środek koła, druga oznacza obramowanie koła. Następnie nanoszony jest tekst, przy pomocy funkcji ImageTTFText()

```
ImageTTFText($map,24,0,55,65,$kolor_czerwony,$czcionka_georgia,"WARD
RIVING & WARCHALKING @ Katowice, Poland");
```

```
ImageTTFText($map,24,0,55,100,$kolor_czerwony,$czcionka_georgia,"I
found $wapkce ($data) Wireless Access Points in Katowice, Poland");

ImageTTFText($map,24,0,55,135,$kolor_czerwony,$czcionka_georgia,"Thi
s work by Adam Michal Ziaja (http://adamziaja.com) is licensed under
a Creative Commons Attribution-ShareAlike 2.0 License");

ImageTTFText($map,24,0,55,170,$kolor_czerwony,$czcionka_georgia,"OPE
N node - green, WEP node - blue, WPA node - red");
```

Liczby oznaczają tutaj pozycję na obrazku znajdującym się w zmiennej $map, kolory oraz czcionki ostały zdefiniowane wcześniej

```
$czcionka_georgia='Georgia.ttf';

$kolor_bialy=ImageColorAllocate($map,255,255,255);
```

Przy nazwach czcionek należy zwrócić uwagę na wielkość znaków, ponieważ system Linux rozróżnia duże oraz małe znaki, kolory zapisywane są w RGB.

6. Uruchamianie i testowanie

Skrypty po stronie użytkownika mają standardowe wymagania jeśli chodzi o zasoby, natomiast skrypty po stronie administratora nie są standardowymi skryptami PHP ze względu na obciążenia jakie generują.

Szczególnie przy skrypcie adm/mysql.php oraz adm/staticmaps.php mogą wystąpić błędy ze względu na ograniczenia nałożone standardowo na PHP, pierwszy skrypt parsuje w locie pliki wielkości kilku MB, natomiast drugi skleja elementy mapy nakładając na nią punkty z bazy danych. W związku tym PHP może brakować pamięci RAM, w tym wypadku wystąpi krytyczny błąd

```
Fatal error: Allowed memory size of X bytes exhausted (tried to
allocate Y bytes)
```

Zmienna memory_limit, która odpowiedzialna jest ilość pamięci RAM, która ma do wykorzystania PHP w wersjach poniżej 5.2.0 ustawiono była standartowo na wartość 8M, w wersji 5.2.0 na 16M, natomiast już w późniejszych wydaniach na 128M. Mimo wszystko wciąż może to być niewystarczająca ilość, przy testowaniu nakładałem lokalnie limity rzędu 512M. Ze względów bezpieczeństwa najlepiej nie modyfikować standardowej wartości z php.ini, a jedynie dopisać linię

```
php_value memory_limit "512M"
```

Do pliku .htaccess znajdującym się w katalogu adm, w którym to przechowywane są skrypty administracyjne.

Kolejnym częstym błędem powyższych skryptów jak i również adm/geocode.php jest też przekroczenie limitu czasu wykonywania skryptu, który standardowo ustawiony jest na 30 sekund, w tym wypadku występuje błąd

```
Fatal error: Maximum execution time of 30 seconds exceeded
```

Odpowiada za to wartość set_time_limit, w celu wyeliminowania błędu na początku skryptów, które tego wymagają dopisana została linia

```
set_time_limit(0)
```

Funkcja ta odpowiedzialna jest za zniesienie limitów czasowych na wykonywanie skryptów.

Jednym z ważnych czynników jest brak możliwości włączenia PHP Safe Mode, który w znacznym stopniu poprawia bezpieczeństwo wykonywania skryptów na serwerze. Głównie z

uwagi na zastosowane w skryptach administracyjnych funkcje shell_exec, jak i również set_time_limit oraz modyfikacje memory_limit. W związku z tym położyłem nacisk na bezpieczeństwo aplikacji web. Przy wszystkich zmiennych pobieranych za pomocą GET do bazy danych zostały zastosowane praktycznie wszystkie możliwe funkcje poprawiające bezpieczeństwo przekazywania zmiennej zewnętrznej do zapytania SQL, czyli

```
mysql_real_escape_string(), htmlspecialchars(), addslashes()
```

Zostały one za każdym razem zastosowane kolejno po sobie

```
mysql_real_escape_string(htmlspecialchars(addslashes($_GET['bssid'])
,ENT_QUOTES));
```

Jednocześnie nie mają odczuwalnego negatywnego pływu na przepływ danych i nie wynikają z tego powodu żadne błędy. Ponadto skrypty, które pobierają zmienną za pomocą GET jednocześnie logują pobierane informacje. Jeśli przykładowo haker będzie próbował wywołać

```
ap.php?bssid=neostrada'%20LIMIT%20100%20%20/*
```

Czyli wstawić ciąg "neostrada' LIMIT 100 /*" do zapytania

```
SELECT * FROM wardriving WHERE bssid = '".$bssid."' LIMIT 1
```

Tym samym wywołać kod

```
SELECT * FROM wardriving WHERE bssid = 'neostrada' LIMIT 100 /*'
LIMIT 1
```

Zamiast jednego pobierze 100 wyników, a dalszą część zapytania zakomentuje, oczywiście to tylko bardzo łatwy przykład metody ataku. Możliwe są bardziej zaawansowane ataki na (My)SQL, które przynoszą o wiele gorsze skutki. Ponadto zapytania zostały tak ułożone, a żeby ograniczyć inne metody ataków, np. poprzez wpisanie znaku %, który odpowiada za każdy ciąg znaków, ponieważ po przez zastosowanie

```
bssid = ''
```

Po wstawieniu znaku % do zapytania

```
bssid = '%'
```

Baza nie zwróci żadnego rekordu (chyba, że sieć będzie nazywać się "%"), jednak w przypadku zapytania

```
bssid like ''
```

i wstawieniu znaku %

```
bssid like '%'
```

Skrypt wyświetliłby wszystkie rekordy z bazy danych. Dodatkowo skrypt odnotuje błąd, w tym wypadku próbę ataku w logach, w następujący sposób

```
| id | ip | date | error | request | string |

| 1 | 127.0.0.1 | 2011-02-02 13:57:48 | y |
/az/ap.php?bssid='%20LIMIT%20100%20%20/* | \' LIMIT 100  /* |
```

Gdzie zostaje odnotowane IP, data, wpisany adres oraz ciąg znaków do zmiennej. Dodatkowo rozwiązanie te pozwala na diagnozowanie błędów w skrypcie, przykładowo BSSID, który nie występuje w bazie danych.

Firma Google udostępnia dane do geokodowania, jednak nakłada znaczne limity na ilość zapytań, jest to 2500 zapytań na dzień, dodatkowo wprowadzone są limity czasowe, które nie są do końca znane, z tego powodu skrypt adm/geocode.php został rozbudowany o zapytania przez sieć TOR w celu zwiększenia limitu zapytań oraz o obsługę błędów, które zwracają serwery Google. Skrypt w przypadku otrzymania statusu OVER_QUERY_LIMIT odczekuje dodatkowe 10 sekund, ponieważ limity czasowe prawdopodobnie zaczynają odliczanie od początku w przypadku ponownej próby w czasie trwania aktualnego limitu. Niestety dokładny algorytm tej usługi nie jest znany.

W przypadku skrypty adm/staticmaps.php również występują problemy z limitami nałożonymi przez serwery firmy Google, z których pobierane są dane w przypadku budowania warstwy mapy bazującej na informacjach z Google Maps. W tym wypadku limity są jednak mniej restrykcyjne, ponieważ skrypt tak samo jak użytkownik przeglądający mapę potrzebuje pobrać większą ilość warstw, jednak przy paru próbach pobrania całej mapy od zera występuje błąd

```
Żądanie HTTP wysłano, oczekiwanie na odpowiedź... 403 Forbidden
```

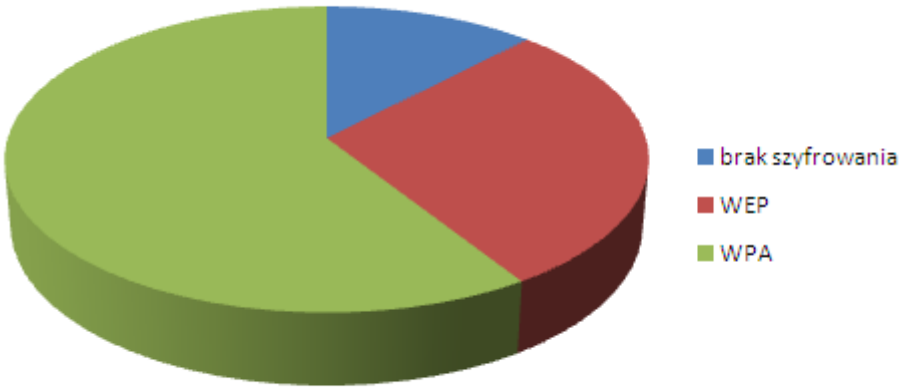
Z tego powodu zostało wprowadzone cache w skrypcie. Odpowiada za to definicja

```
MAP_TILES_CACHE_TIME
```

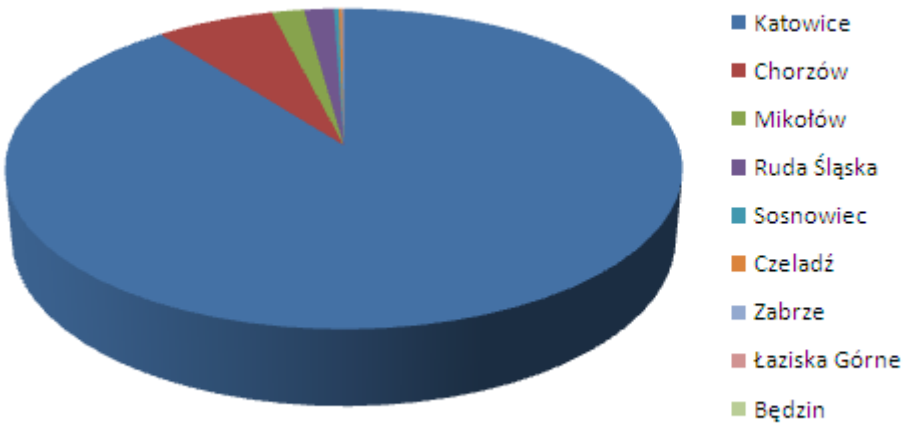
Przyjmuje ona ilość dni przez które będzie trzymana każda warstwa mapy, pozwala to również na budowanie mapy z aktualnymi wynikami, gdyż z czasem informacje mogą zostać rozbudowane lub poprawione.

7. Analiza wyników

Na dzień 6 lutego 2011 łącznie znalazłem 12263 sieci bezprzewodowych, z czego 1491 (12.159%) było bez szyfrowania, 3530 (28.786%) z szyfrowaniem WEP, 7242 (59.056%) z szyfrowaniem WPA.



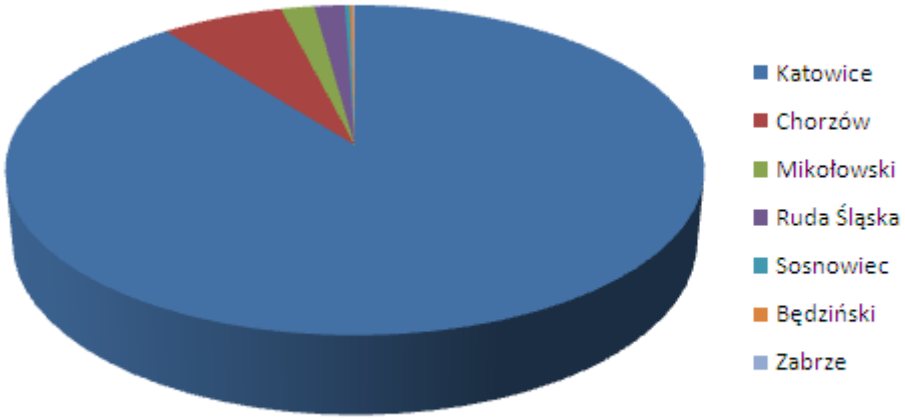
Oznacza to, że co najmniej ponad 40% sieci nie jest zabezpieczona, z uwagi na zastosowany sposób szyfrowania, lub w ogóle jego brak. Szyfrowanie WEP nie jest bezpieczne, ponieważ jego łamanie sprowadza się głównie do czasu zbierania wektorów inicjalizujących (IV), 40-bitowy WEP (64 bitowy klucz) może zostać złamany z 300,000 IV, 104-bitowy WEP (128 bitowy klucz) może zostać złamany z 1,500,000 IVs, przytoczone wartości są wartościami orientacyjnymi. Przy zastosowaniu wstrzykiwania pakietów czas ten ulega znacznemu skróceniu ze względu na wzmożony przepływ IV. Po zebraniu pakietów samo łamanie hasła trwa na dzisiejszych komputerach od niecałej sekundy do paru minut. W związku z tym jedynym bezpiecznym sposobem szyfrowania jest na dzień dzisiejszy WPA.



Miasta wg ilości występowania:

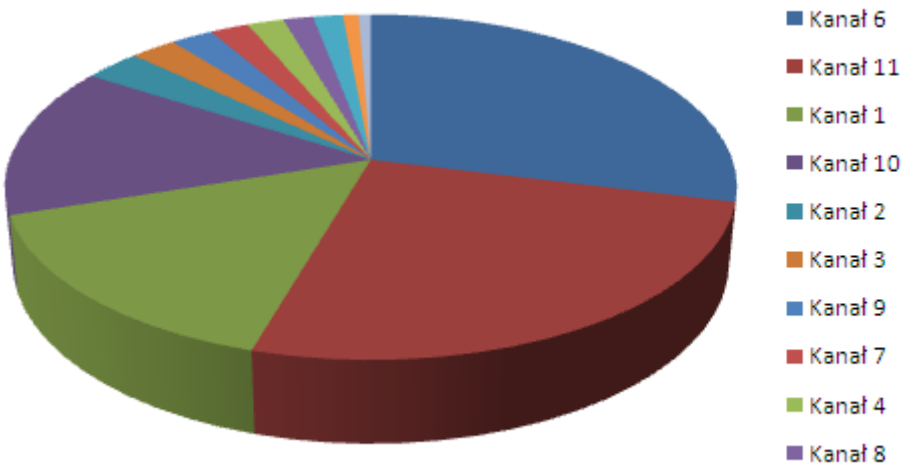
1. Katowice - 10977 (89.51%)
2. Chorzów - 801 (6.53%)
3. Mikołów - 218 (1.78%)
4. Ruda Śląska - 200 (1.63%)
5. Sosnowiec - 33 (0.27%)
6. Czeladź - 22 (0.18%)
7. Zabrze - 9 (0.07%)
8. Łaziska Górne - 2 (0.02%)
9. Będzin - 1 (0.01%)

Głównym obszarem badań były Katowice, z tego powodu aż 89.5% sieci zostało znalezionych w Katowicach.



Powiaty wg ilości występowania:

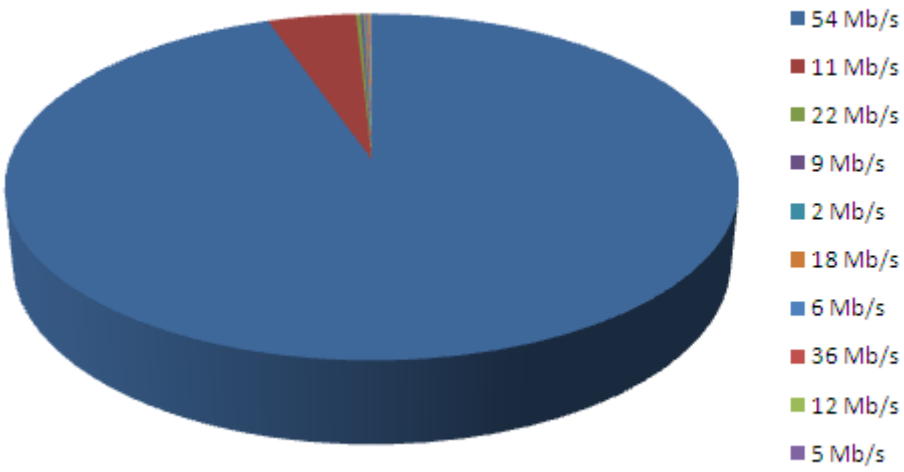
1. Katowice - 10977 (89.51%)
2. Chorzów - 801 (6.53%)
3. Mikołowski - 220 (1.79%)
4. Ruda Śląska - 200 (1.63%)
5. Sosnowiec - 33 (0.27%)
6. Będziński - 23 (0.19%)
7. Zabrze - 9 (0.07%)



Kanały wg ilości występowania:

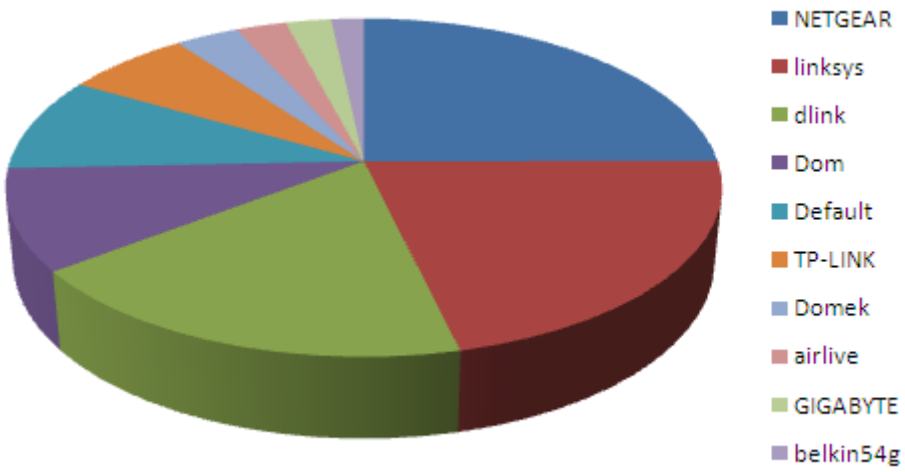
1. Kanał 6 - 3541 (28.875%)
2. Kanał 11 - 3138 (25.589%)
3. Kanał 1 - 1888 (15.396%)
4. Kanał 10 - 1737 (14.165%)
5. Kanał 2 - 362 (2.952%)
6. Kanał 3 - 292 (2.381%)
7. Kanał 9 - 268 (2.185%)
8. Kanał 7 - 244 (1.99%)
9. Kanał 4 - 231 (1.884%)
10. Kanał 8 - 190 (1.549%)
11. Kanał 5 - 187 (1.525%)
12. Kanał 13 - 99 (0.807%)
13. Kanał 12 - 77 (0.628%)

Jak widać aż 84% sieci stosuje prawdopodobnie standardowy kanał przydzielony do routera, którym to najczęściej jest kanał 1, 6, 10 lub 11. Związane jest to bezpośrednio wzajemnym zakłócaniem się kanałów, ponieważ każdy kanał zakłóca 5 kanałów w górę i w dół, dlatego jedynie 1, 6 oraz 11 kanał nie zachodzą na siebie, w związku z tym nie zakłócają się.



Prędkość:

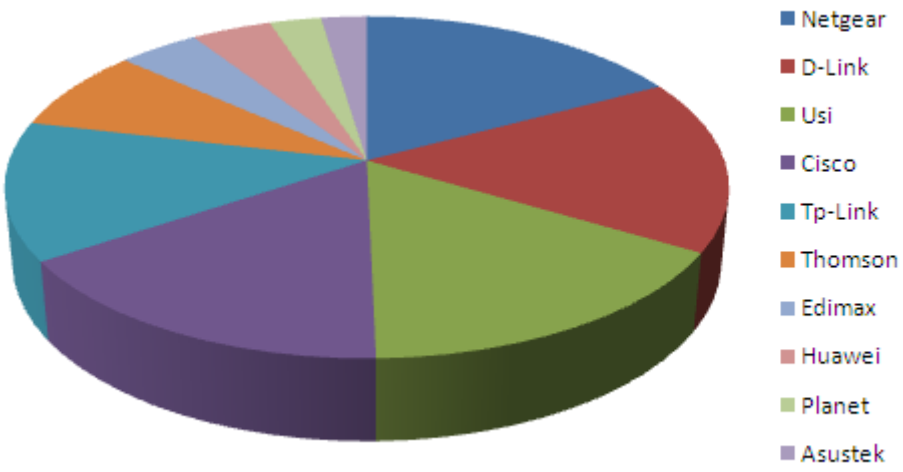
1. 54 Mb/s - 11605 (94.634%)
2. 11 Mb/s - 558 (4.55%)
3. 22 Mb/s - 28 (0.228%)
4. 9 Mb/s - 14 (0.114%)
5. 2 Mb/s - 10 (0.082%)
6. 18 Mb/s - 10 (0.082%)
7. 6 Mb/s - 9 (0.073%)
8. 36 Mb/s - 8 (0.065%)
9. 12 Mb/s - 6 (0.049%)
10. 5 Mb/s - 6 (0.049%)
11. 24 Mb/s - 5 (0.041%)
12. 1 Mb/s - 2 (0.016%)
13. 48 Mb/s - 1 (0.008%)



Dziesięć najczęściej występujących SSID:

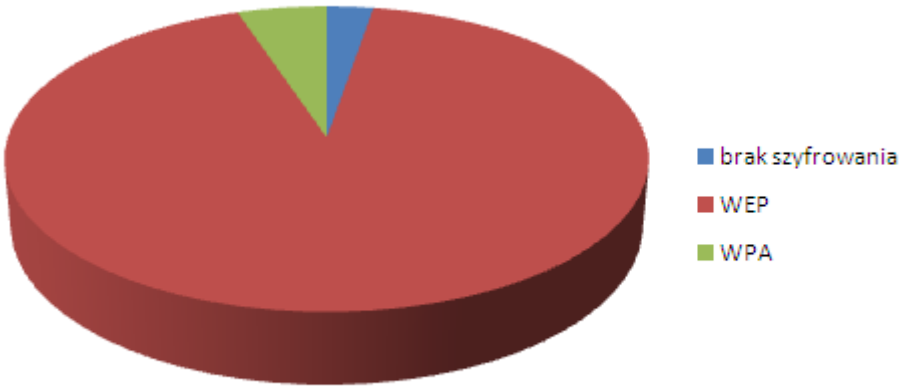
1. NETGEAR - 396 (3.229%)
2. linksys - 338 (2.756%)
3. dlink - 291 (2.373%)
4. Dom - 154 (1.256%)
5. Default - 143 (1.166%)
6. TP-LINK - 104 (0.848%)
7. Domek - 53 (0.432%)
8. airlive - 42 (0.342%)
9. GIGABYTE - 38 (0.31%)
10. belkin54g - 27 (0.22%)

Jak widać najczęściej użytkownicy nie zmieniają standardowej nazwy urządzenia, może to prowadzić do problemów z połączeniem, kiedy użytkownik będzie próbował ręcznie podłączyć się do swojej sieci, a w efekcie będzie podłączał się do innej sieci o identycznej nazwie.



Dziesięć najczęściej występujących producentów:

1. Netgear - 1635 (13.333%)
2. D-Link - 1571 (12.811%)
3. Usi - 1541 (12.566%)
4. Cisco - 1527 (12.452%)
5. Tp-Link - 1254 (10.226%)
6. Thomson - 746 (6.083%)
7. Edimax - 404 (3.294%)
8. Huawei - 398 (3.246%)
9. Planet - 257 (2.096%)
10. Asustek - 227 (1.851%)

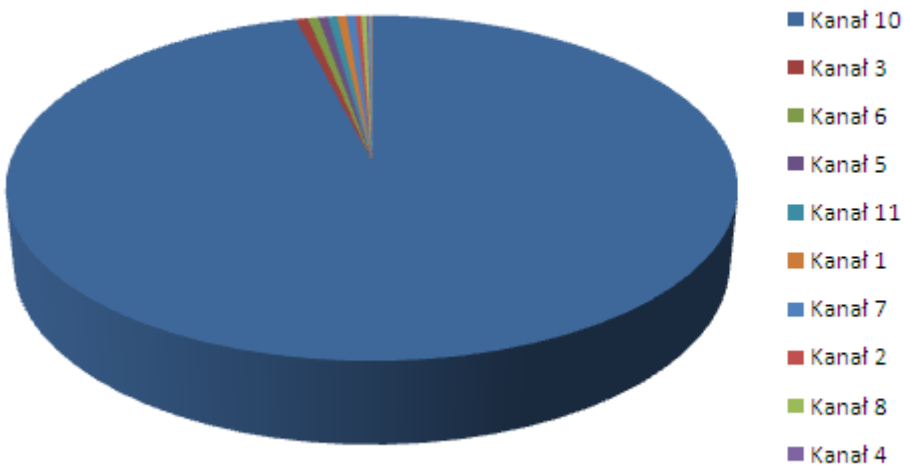


Ilość Neostrad ze standardowym SSID: 1458

Ilość Neostrad bez szyfrowania: 39 (2.675%)

Ilość Neostrad z szyfrowaniem WEP: 1344 (92.181%)

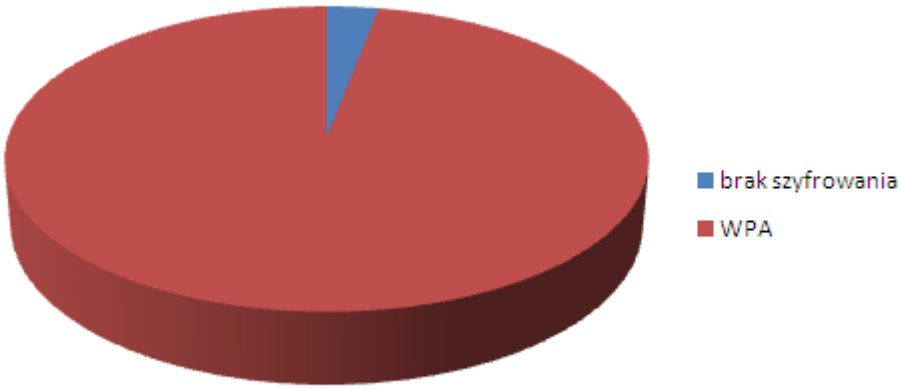
Ilość Neostrad z szyfrowaniem WPA: 75 (5.144%)



Ilość Neostrad występujących na poszczególnych kanałach:

1. Kanał 10 - 1401 (96.091%)
2. Kanał 3 - 9 (0.617%)
3. Kanał 6 - 8 (0.549%)

4. Kanał 5 - 7 (0.48%)
5. Kanał 11 - 7 (0.48%)
6. Kanał 1 - 7 (0.48%)
7. Kanał 7 - 7 (0.48%)
8. Kanał 2 - 4 (0.274%)
9. Kanał 8 - 4 (0.274%)
10. Kanał 4 - 2 (0.137%)
11. Kanał 9 - 1 (0.069%)
12. Kanał 12 - 1 (0.069%)

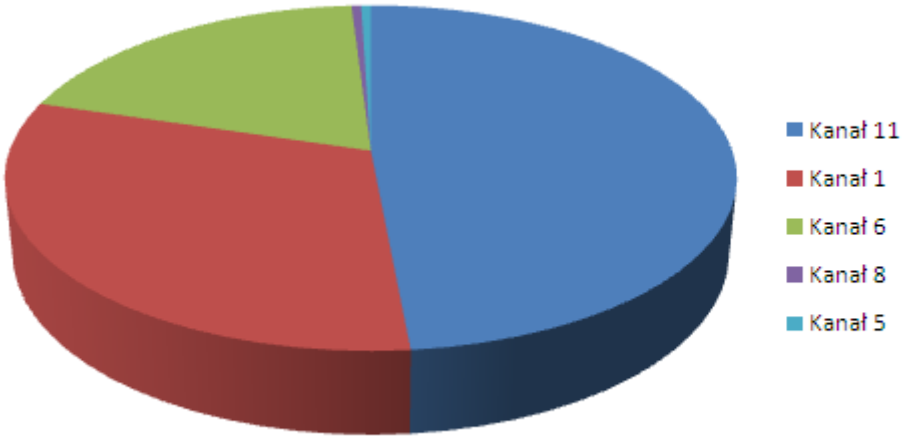


Ilość Netii ze standardowym SSID: 202

Ilość Netii bez szyfrowania: 6 (2.97%)

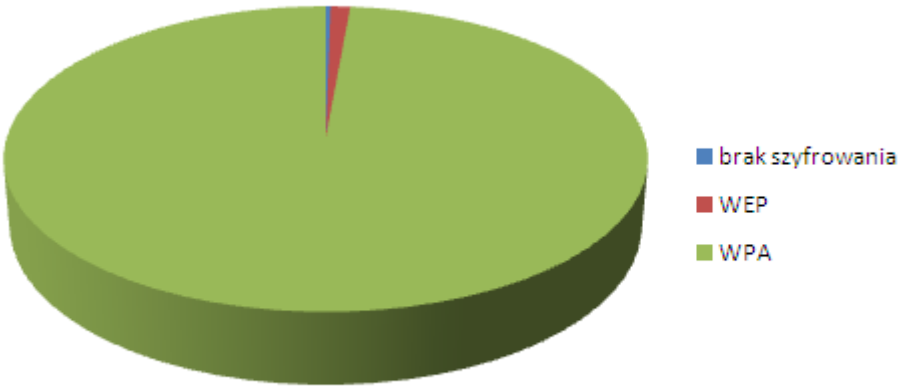
Ilość Netii z szyfrowaniem WEP: 0 (0%)

Ilość Netii z szyfrowaniem WPA: 196 (97.03%)



Ilość Netii występujących na poszczególnych kanałach:

1. Kanał 11 - 98 (48.515%)
2. Kanał 1 - 63 (31.188%)
3. Kanał 6 - 39 (19.307%)
4. Kanał 8 - 1 (0.495%)
5. Kanał 5 - 1 (0.495%)

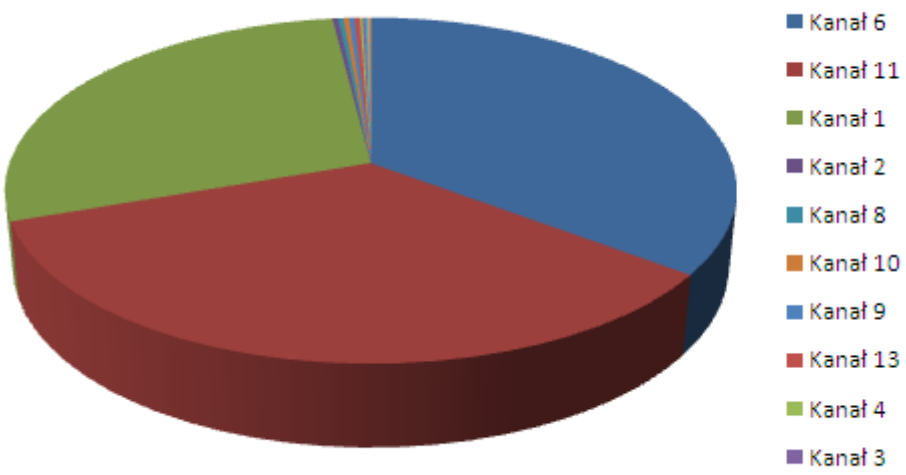


Ilość UPC ze standardowym SSID: 712

Ilość UPC bez szyfrowania: 2 (0.281%)

Ilość UPC z szyfrowaniem WEP: 8 (1.124%)

Ilość UPC z szyfrowaniem WPA: 702 (98.596%)



Ilość UPC występujących na poszczególnych kanałach:

1. Kanał 6 - 251 (35.253%)
2. Kanał 11 - 245 (34.41%)
3. Kanał 1 - 202 (28.371%)
4. Kanał 2 - 2 (0.281%)
5. Kanał 8 - 2 (0.281%)
6. Kanał 10 - 2 (0.281%)
7. Kanał 9 - 2 (0.281%)
8. Kanał 13 - 2 (0.281%)
9. Kanał 4 - 1 (0.14%)
10. Kanał 3 - 1 (0.14%)
11. Kanał 12 - 1 (0.14%)
12. Kanał 7 - 1 (0.14%)

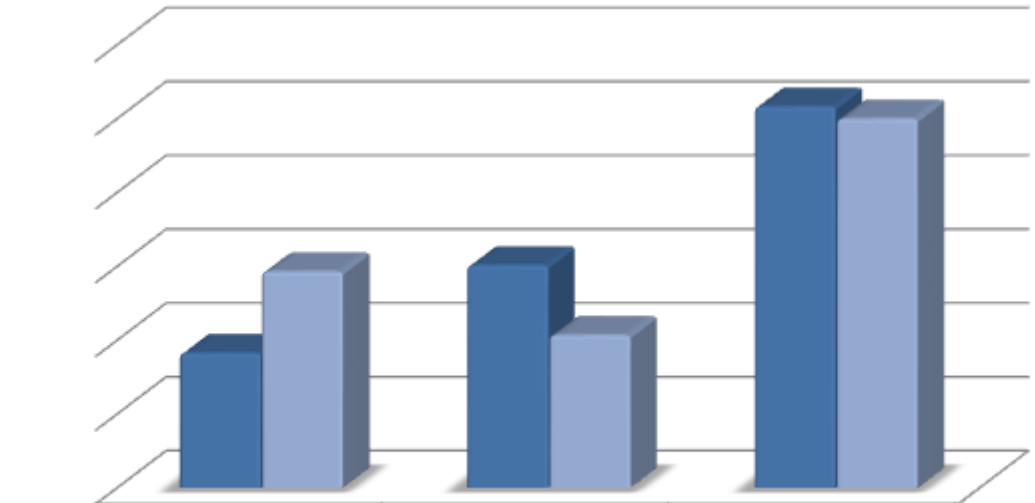
Z powyższych statystyk Neostrady, Netii oraz UPC można wywnioskować, że standardowo najniebezpieczniejszą usługą jest usługa Neostrada Livebox TP. Wiąże się to bezpośrednio z faktem, że użytkownik dostaje prekonfigurowany modem oraz hasło do kluczu WEP, w związku z tym aż 92% użytkowników usługi z pośród znalezionych 1458 stosuje standardowe zabezpieczenie, a raptem 75, czyli 5% stosuje bezpieczny sposób

szyfrowania czyli WPA. Pozostali dwaj dostawcy z pośród najpopularniejszych standardowo stosują szyfrowanie WPA, w związku z tym 97% użytkowników Netii stosowało WPA, natomiast w UPC 98% użytkowników stosowało WPA.

8. Podsumowanie

Sposób badań sieci jaki zastosowałem jest praktycznie jedynym sposobem na masowe zbadanie sieci na danym obszarze, dane są stosunkowo dokładne ponieważ sama geolokalizacja pozycji GPS na których zostały znalezione sieci bazuje na danych firmy Google, natomiast pozostałe informacje wynikają bezpośrednio z wykrytych pakietów przez program Kismet, który to jest niewykrywalny jako pasywny skaner sieci bezprzewodowych, dlatego też istnieje bardzo znikome prawdopodobieństwo, że którakolwiek sieć wysyłała fałszywe pakiety.

Podobny audyt bezpieczeństwa sieci bezprzewodowych w Katowicach wykonała firma Kaspersky w latach 2009, 2010 i zaprezentowała wyniki badań pod adresem http://www.kaspersky.pl/about.html?s=news_reviews&cat=3&newsid=1412.



	WEP	Bez szyfrowania	WPA/WPA2
2010	18,24%	30,13%	51,63%
2009	29,3%	20,7%	50,0%

Źródło: http://www.kaspersky.pl

Jak widać wyniki badań firmy Kaspersky dość odbiegają od wyników moich badań co jest w dużej mierze związane z ilością znalezionych sieci, gdyż w moim przypadku było ich prawie 10 razy tyle, a sam okres badania obejmował mniej więcej ten sam okres czasu, czyli rok 2010, ponieważ pierwsza siec w mojej bazie danych pochodzi z dnia 2010-03-27, a ostatnia z 2011-01-02 (na dzień 6 lutego 2011).

Dodatkowo wykryłem 9 sieci działających na nielegalnym w Polsce 14 kanale. Ponieważ w Polsce tylko częstotliwości od 2,4000 do 2,4835 GHz nie wymagają koncesji, a

14 kanał działa na częstotliwości 2,484 GHz. Owe sieci można otrzymać po przez przykładowe zapytanie do bazy MySQL

```
SELECT bssid,ssid,gpslat,gpslon FROM wardriving WHERE channel = 14
```

które zwraca sieci na 14 kanale

```
| bssid | ssid | gpslat | gpslon |
| 00:02:6F:3F:44:F1 | TransX300 | 50.192749 | 18.98068 |
| 00:02:6F:54:0F:EC | Trans_FijPn | 50.1939 | 18.979565 |
| 00:02:6F:3F:39:98 | TransPd1 | 50.183872 | 19.007647 |
| 00:02:6F:43:29:D6 | Trans-Net2 | 50.183879 | 19.008851 |
| 00:02:6F:46:F2:DD | TransKost | 50.184553 | 19.008068 |
| 00:02:6F:46:F3:39 | cyfranet4 | 50.183893 | 19.008741 |
| 00:02:6F:4B:E6:0B | TransJarr | 50.183939 | 19.0093 |
| 00:90:4C:75:04:14 | IT-Wic(9) 0322599523 | 50.26511 | 19.009583 |
| 00:90:CC:CD:EC:FC | u_maniaka | 50.267174 | 19.00998 |
```

Tego typu dane, które zawarte są w bazie danych projektu wykorzystywane są m.in. przez Współrzędne Google (od ang. Google Latitude) do określania położenia przy pomocy Internetu oraz urządzenia Wi-Fi wbudowanego w np. telefon komórkowy, przy czym bez urządzenia GPS. Aplikacja Współrzędne Google pobiera informacje o dostępnych sieciach bezprzewodowych oraz sygnale z jakim są wykrywane, jak również informacje o stacjach bazowych sieci GSM, następnie wysyła zebrane informacje do serwerów firmy Google, które zwracają przybliżoną pozycję GPS na podstawie tych danych. Istnieje więc możliwość stworzenia podobnej aplikacji, która na terenie Katowic powinna mieć o wiele większą dokładność z uwagi na ilość znalezionych sieci.

O ile czas pozwoli, mam zamiar sukcesywnie rozwijać projekt na mojej stronie internetowej znajdującej się pod adresem http://adamziaja.com.

9. Bibliografia

- http://www.dis.org/shipley/ Peter Shipley [dostęp 6 luty 2011]
- http://www.blackbeltjones.com/warchalking/ Warchalking [dostęp 6 luty 2011]
- http://www.kismetwireless.net Kismet [dostęp 6 luty 2011]
- http://dev.openlayers.org OpenLayers Developer Documentation [dostęp 6 luty 2011]
- http://wiki.openstreetmap.org OpenStreetMap Wiki [dostęp 6 luty 2011]
- http://pl.wikipedia.org Wikipedia, wolna encyklopedia [dostęp 6 luty 2011]
- http://pl.wikipedia.org/wiki/IEEE_802.11 IEEE 802.11 – Wikipedia [dostęp 6 luty 2011]
- http://pl.wikipedia.org/wiki/Wi-Fi Wi-Fi – Wikipedia [dostęp 6 luty 2011]
- http://pl.wikipedia.org/wiki/Bezprzewodowa_sie%C4%87_lokalna Bezprzewodowa sieć lokalna – Wikipedia [dostęp 6 luty 2011]
- http://en.wikipedia.org Wikipedia, wolna encyklopedia [dostęp 6 luty 2011]
- http://en.wikipedia.org/wiki/Wardriving Wardriving – Wikipedia [dostęp 6 luty 2011]
- http://en.wikipedia.org/wiki/Warchalking Warchalking – Wikipedia [dostęp 6 luty 2011]
- ftp://ftp.helion.pl/online/siebp2/siebp2-3.pdf Przewodnik po sieciach Wi-Fi i szerokopasmowych sieciach bezprzewodowych [dostęp 6 luty 2011]
- http://www.aircrack-ng.org Aircrack-ng [dostęp 6 luty 2011]
- http://code.google.com/intl/pl/apis/maps/documentation/geocoding/ The Google Geocoding API [dostęp 6 luty 2011]

10. Zawartość płyty CD

- Praca w formacie DOCX
- Praca w formacie PDF
- Kody źródłowe